

**APLIKASI BIMBINGAN SKRIPSI *ONLINE*
MAHASISWA JURUSAN PENDIDIKAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI YOGYAKARTA**

SKRIPSI

Diajukan kepada Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Yogyakarta
untuk Memenuhi Sebagian Persyaratan
guna Memperoleh Gelar Sarjana Sains



Oleh:

Muhammad Jaka P.

NIM 10305141032

**PROGRAM STUDI MATEMATIKA
JURUSAN PENDIDIKAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI YOGYAKARTA
2014**

PERSETUJUAN

Skripsi yang berjudul:

APLIKASI BIMBINGAN SKRIPSI *ONLINE*
MAHASISWA JURUSAN PENDIDIKAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI YOGYAKARTA

Oleh:

Muhammad Jaka P.

NIM 10305141032



Yogyakarta, Juli 2014
Dosen Pembimbing

Nur Hadi Waryanto M.Eng
NIP.19780119 200312 1 002

PERNYATAAN

Yang bertanda tangan dibawah ini, saya:

Nama : Muhammad Jaka P.

NIM : 10305141032

Program Studi : Matematika

Fakultas : Matematika dan Ilmu Pengetahuan Alam

Judul Skripsi : APLIKASI BIMBINGAN SKRIPSI *ONLINE* MAHASISWA
JURUSAN PENDIDIKAN MATEMATIKA FAKULTAS
MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI YOGYAKARTA

Menyatakan bahwa skripsi ini benar-benar karya saya sendiri dan sepanjang sepengetahuan saya, tidak terdapat karya atau pendapat yang ditulis atau diterbitkan orang lain, kecuali pada bagian-bagian tertentu yang diambil sebagai acuan atau kutipan dengan mengikuti tata penulisan karya ilmiah yang telah lazim.

Apabila terbukti pernyataan saya ini tidak benar, maka sepenuhnya menjadi tanggung jawab saya, dan saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Yogyakarta, Juli 2014
Yang Menyatakan



Muhammad Jaka P.
NIM. 10305141032

PENGESAHAN

Skripsi yang berjudul
**“APLIKASI BIMBINGAN SKRIPSI *ONLINE* MAHASISWA JURUSAN
PENDIDIKAN MATEMATIKA FAKULTAS MATEMATIKA DAN ILMU
PENGETAHUAN ALAM UNIVERSITAS NEGERI YOGYAKARTA”**

Yang disusun oleh
Nama : Muhammad Jaka P.
NIM : 10305141032
Prodi : Matematika

Skripsi ini telah diujikan di depan Dewan Penguji pada tanggal 14 Juli 2014 dan
dinyatakan **LULUS**

DEWAN PENGUJI			
Nama	Jabatan	Tanda tangan	Tanggal
<u>Nur Hadi Waryanto, M.Eng.</u> NIP. 19780119 200312 1 002	Ketua Penguji		18-07-2014
<u>Husna Arifah, M. Sc.</u> NIP. 19781015 200212 2 001	Sekretaris Penguji		18-07-2014
<u>Dr. Sugiman</u> NIP. 19650228 199101 1 001	Penguji Utama		15-07-2014
<u>Bambang S.H.M., M. Kom.</u> NIP. 19680210 198812 1 001	Penguji Pendamping		18-07-2014

Yogyakarta, 21 Juli 2014

Fakultas Matematika dan Ilmu
Pengetahuan Alam

Dekan



Dr. Hartono

NIP. 19620329 198702 1 002

MOTTO

*Sebuah Generasi lahir untuk melampaui Dinding yang disebut Generasi
Sebelumnya, kemudian Generasi itu akan tumbuh untuk menjadi Dinding
yang akan dilampaui oleh Generasi Selanjutnya*

*Seseorang diberikan kelebihan dari orang lain bukan untuk memenuhi
ambisi pribadi, tetapi untuk menjalankan amanah besar yang diserahkan
kepadanya*

APLIKASI BIMBINGAN SKRIPSI *ONLINE*
MAHASISWA JURUSAN PENDIDIKAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI YOGYAKARTA

Oleh
Muhammad Jaka P.
NIM 10305141032

ABSTRAK

Proses bimbingan tugas akhir skripsi di Jurusan Pendidikan Matematika Universitas Negeri Yogyakarta (UNY) memiliki beberapa kendala yang menyebabkan proses pengerjaan tugas akhir mahasiswa terhambat, yaitu dokumentasi judul skripsi yang kurang tertata dan kesulitan mahasiswa dan dosen untuk mengadakan pertemuan. Untuk mengatasi permasalahan tersebut, perlu dirancang sistem pendukung bimbingan skripsi dengan menerapkan teknologi informasi sehingga dapat bekerja layaknya bimbingan skripsi secara tatap muka.

Sistem yang dirancang dalam penelitian ini dibuat dalam ruang lingkup Jurdikmat UNY berdasarkan informasi dan prosedur pembuatan tugas akhir skripsi Jurusan Pendidikan Matematika UNY yang dapat dilakukan secara *online* dan dirancang menggunakan model *Waterfall*. Aplikasi dirancang dengan bahasa pemrograman PHP dibantu dengan *framework* Yii, dan *Database Management System* (DBMS) MySQL.

Aplikasi Bimbingan Skripsi *Online* yang dirancang terdiri dari 5 tipe pengguna dan 10 pilihan menu. Tipe pengguna yang ada dalam aplikasi ini adalah Administrator, Dosen Pembimbing Skripsi, Ketua Program Studi, Staf Jurusan, dan Mahasiswa Jurdikmat UNY. Pilihan menu yang tersedia adalah Judul, *Upload* Proposal, List Proposal, Dosen, Hak Akses, Pesan, Pengguna, Daftar Bimbingan dan Profil. Berdasarkan hasil pengujian *betha* kepada pengguna dan ahli komputer, aplikasi yang telah dirancang dapat mempermudah pencarian judul skripsi, pengajuan proposal, dan proses bimbingan.

Kata kunci: tugas akhir, sistem, data, aplikasi, online.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, yang telah melimpahkan segala karunia, rahmat dan hidayah-Nya sehingga tugas akhir skripsi ini dapat terselesaikan dengan baik dan lancar dengan bantuan dari berbagai pihak. Oleh karena itu dalam penulis mengucapkan terima kasih kepada:

1. Bapak Dr. Hartono, selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta.
2. Bapak Dr. Agus Maman Abadi selaku Ketua Program Studi Matematika dan Penasehat Akademik.
3. Bapak Nur Hadi Waryanto, M. Eng selaku dosen pembimbing.
4. Teman-teman organisasi HIMATIKA 2011, BEM FMIPA 2012, dan BEM KM UNY 2013 yang telah berbagi banyak pengalaman.
5. Seluruh Mahasiswa Matematika UNY Angkatan 2010 serta semua pihak yang telah membantu secara langsung maupun tidak langsung.

Dalam pembuatan skripsi ini masih banyak kekurangan. Oleh karena itu, kritik dan saran sangat diharapkan sebagai koreksi. Semoga skripsi ini bermanfaat bagi penulis dan akademisi yang tertarik dengan aplikasi *online* khususnya aplikasi yang dibuat dengan *framework* Yii.

Yogyakarta, Juli 2014

Penulis



Muhammad Jaka P.

NIM. 10305141032

DAFTAR ISI

Halaman

JUDUL	i
PERSETUJUAN	ii
PERNYATAAN.....	iii
PENGESAHAN	iv
MOTTO	v
ABSTRAK	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
BAB I.....	1
A. Latar Belakang	1
B. Batasan Masalah.....	3
C. Rumusan Masalah	3
D. Tujuan Penelitian	4
E. Manfaat Penelitian	4
BAB II.....	5
A. Deskripsi Teori.....	5
1. Tugas Akhir di UNY	5
2. Sistem Informasi.....	8
3. Basis Data.....	10
4. Model Basis Data Relasional	11
5. Database Management Sistem (DBMS).....	14

6.	Model Perancangan Sistem	14
7.	<i>Flowmap</i>	16
8.	<i>Data Flow Diagram</i> (DFD).....	17
9.	PHP.....	21
10.	MySQL	22
11.	<i>Framework Yii</i>	24
12.	<i>Software Requirements Specification</i> (SRS).....	29
13.	Jaminan Kualitas Perangkat Lunak.....	31
B.	Kerangka Berpikir.....	33
BAB III	35
A.	Analisis Sistem.....	35
1.	Analisis Masalah	35
2.	Analisis Prosedur yang Terlibat	36
3.	Analisis Dokumen yang Digunakan.....	39
4.	Analisis Kebutuhan	40
B.	Perancangan Sistem	44
1.	Perancangan Basis Data	44
2.	Perancangan Aliran Data.....	50
3.	Perancangan Struktur Menu	64
4.	Perancangan Antar Muka	68
C.	Implementasi Sistem	100
1.	Implementasi Perangkat Keras	101
2.	Implementasi Perangkat Lunak	101
3.	Implementasi Antar Muka.....	101
D.	Pengujian Sistem.....	128

BAB IV	136
A. Kesimpulan	136
B. Saran.....	137
DAFTAR PUSTAKA	139

DAFTAR TABEL

Tabel 1. Dokumen Yang Digunakan Dalam Pembuatan Skripsi	40
Tabel 2. Struktur Tabel authassgment	45
Tabel 3. Struktur Tabel authitem	45
Tabel 4. Struktur Tabel authitemchild	46
Tabel 5. Struktur Tabel dosen	46
Tabel 6. Struktur Tabel mahasiswa	47
Tabel 7. Struktur Tabel proposal	47
Tabel 8. Struktur Tabel mailbox conversation	48
Tabel 9. Struktur Tabel mailbox message	48
Tabel 10. Struktur Tabel users	49
Tabel 11. Persentase Hasil Pengujian Betha Kepada Pengguna	131
Tabel 12. Persentase Hasil Pengujian Kepada Ahli Komputer	133
Tabel 13. Keterangan Kelayakan, Saran, Dan Revisi Aplikasi	134

DAFTAR GAMBAR

Gambar 1. Alur Pendaftaran Skripsi	7
Gambar 2. Alur Pendaftaran Ujian Skripsi	8
Gambar 3. Model Waterfall Presmann	15
Gambar 4. Keterangan Gambar Dalam Flowmap	16
Gambar 5. Contoh Flowmap	17
Gambar 6. Keterangan Gambar Dalam DFD	19
Gambar 7. Contoh Diagram Konteks	20
Gambar 8. Contoh DFD Level 1	20
Gambar 9. Contoh DFD Level 2	20
Gambar 10. Aplikasi Yii	25
Gambar 11. Model Generator Yii	26
Gambar 12. Crud Generator Yii	27
Gambar 13. Controller Generator Yii	28
Gambar 14. Flowmap Prosedur Mengajukan Judul Skripsi	37
Gambar 15. Flowmap Prosedur Bimbingan Skripsi	38
Gambar 16. Flowmap Prosedur Pendaftaran Ujian Tugas Akhir Skripsi	39
Gambar 17. Skema Relasi	50
Gambar 18. Diagram Konteks Sistem Informasi Bimbingan Skripsi Online	51
Gambar 19. DFD Level 1 Sistem Informasi Bimbingan Skripsi Online	52
Gambar 20. DFD Level 2 Proses 1.0 (<i>Login</i>)	54
Gambar 21. DFD Level 2 Proses 2.0 (Data Pengguna)	55
Gambar 22. DFD Level 2 Proses 3.0 (Data Pesan)	57
Gambar 23. DFD Level 2 Proses 4.0 (Data Dosen)	58
Gambar 24. DFD Level 2 Proses 5.0 (Data Proposal)	59
Gambar 25. DFD Level 2 Proses 6.0 (Data Mahasiswa)	60
Gambar 26. DFD Level 3 Proses 2.2 (Profil Pengguna)	62
Gambar 27. DFD Level 3 Proses 2.3 (Hak akses Pengguna)	63
Gambar 28. Struktur Menu Administrator	65
Gambar 29. Struktur Menu Dosen	66
Gambar 30. Struktur Menu Kaprodi	67

Gambar 31. Struktur Menu Mahasiswa	67
Gambar 32. Struktur Menu Staf.....	68
Gambar 33. Rancangan Antar Muka Halaman Login.....	69
Gambar 34. Rancangan Antar Muka Menu Pengguna (Administrator).....	70
Gambar 35. Rancangan Antar Muka Menu Pengguna (Dosen).....	70
Gambar 36. Rancangan Antar Muka Menu Pengguna (Kaprodi).....	71
Gambar 37. Rancangan Antar Muka Menu Pengguna (Mahasiswa).....	72
Gambar 38. Rancangan Antar Muka Menu Pengguna (Staf)	72
Gambar 39. Rancangan Antar Muka Menu Judul (Administrator).....	73
Gambar 40. Rancangan Antar Muka Menu Judul (Dosen).....	73
Gambar 41. Rancangan Antar Muka Menu Judul (Kaprodi).....	74
Gambar 42. Rancangan Antar Muka Menu Judul (Mahasiswa)	74
Gambar 43. Rancangan Antar Muka Menu Judul (Staf).....	75
Gambar 44. Rancangan Antar Muka Form Tambah Data Judul (Administrator).....	75
Gambar 45. Rancangan Antar Muka Form Tambah Data Judul (Staf).....	76
Gambar 46. Rancangan Antar Muka Detail Data Judul (Administrator).....	76
Gambar 47. Rancangan Antar Muka Detail Data Judul (Dosen).....	77
Gambar 48. Rancangan Antar Muka Detail Data Judul (Kaprodi)	77
Gambar 49. Rancangan Antar Muka Detail Data Judul (Mahasiswa)	78
Gambar 50. Rancangan Antar Muka Detail Data Judul (Staf).....	78
Gambar 51. Rancangan Antar Muka Menu Upload Proposal (Administrator).....	79
Gambar 52. Rancangan Antar Muka Menu Upload Proposal (Mahasiswa).....	79
Gambar 53. Rancangan Antar Muka Menu List Proposal (Administrator)	80
Gambar 54. Rancangan Antar Muka Menu List Proposal (Kaprodi)	80
Gambar 55. Rancangan Antar Muka Daftar Dosen (Administrator)	81
Gambar 56. Rancangan Antar Muka Daftar Dosen (Dosen)	81
Gambar 57. Rancangan Antar Muka Daftar Dosen (Kaprodi)	82
Gambar 58. Rancangan Antar Muka Daftar Dosen (Mahasiswa).....	82
Gambar 59. Rancangan Antar Muka Daftar Dosen (Staf)	83
Gambar 60. Rancangan Antar Muka Detail Data Dosen (Administrator)	83
Gambar 61. Rancangan Antar Muka Detail Data Dosen (Dosen)	84
Gambar 62. Rancangan Antar Muka Detail Data Dosen (Kaprodi)	84

Gambar 63. Rancangan Antar Muka Detail Data Dosen (Mahasiswa).....	85
Gambar 64. Rancangan Antar Muka Detail Data Dosen (Staf)	85
Gambar 65. Rancangan Antar Muka Menu Hak Akses (Administrator)	86
Gambar 66. Rancangan Antar Muka Menu Hak Akses (Staf).....	86
Gambar 67. Rancangan Antar Muka Daftar Hak Akses (Administrator).....	87
Gambar 68. Rancangan Antar Muka Daftar Hak Akses (Staf).....	87
Gambar 69. Rancangan Antar Muka Menu Pesan (Administrator)	88
Gambar 70. Rancangan Antar Muka Menu Pesan (Dosen)	88
Gambar 71. Rancangan Antar Muka Menu Pesan (Kaprodi)	89
Gambar 72. Rancangan Antar Muka Menu Pesan (Mahasiswa)	89
Gambar 73. Rancangan Antar Muka Menu Pesan (Staf).....	90
Gambar 74. Rancangan Antar Muka Buat Pesan (Administrator).....	90
Gambar 75. Rancangan Antar Muka Buat Pesan (Dosen).....	91
Gambar 76. Rancangan Antar Muka Buat Pesan (Kaprodi)	91
Gambar 77. Rancangan Antar Muka Buat Pesan (Mahasiswa)	92
Gambar 78. Rancangan Antar Muka Buat Pesan (Staf).....	92
Gambar 79. Rancangan Antar Muka Baca Pesan (Administrator)	93
Gambar 80. Rancangan Antar Muka Baca Pesan (Dosen)	93
Gambar 81. Rancangan Antar Muka Baca Pesan (Kaprodi).....	94
Gambar 82. Rancangan Antar Muka Baca Pesan (Mahasiswa).....	94
Gambar 83. Rancangan Antar Muka Baca Pesan (Staf)	95
Gambar 84. Rancangan Antar Muka Menu Users (Administrator)	95
Gambar 85. Rancangan Antar Muka Menu Users (Staf)	96
Gambar 86. Rancangan Antar Muka Form Tambah User (Administrator)	96
Gambar 87. Rancangan Antar Muka Form Tambah User (Staf)	97
Gambar 88. Rancangan Antar Muka Detail User (Administrator)	97
Gambar 89. Rancangan Antar Muka Detail User (Staf)	98
Gambar 90. Rancangan Antar Muka Profil (Administrator)	98
Gambar 91. Rancangan Antar Muka Profil (Dosen).....	99
Gambar 92. Rancangan Antar Muka Profil (Kaprodi).....	99
Gambar 93. Rancangan Antar Muka Profil (Mahasiswa).....	100
Gambar 94. Rancangan Antar Muka Profil (Staf)	100

Gambar 95. Tampilan Antar Muka Halaman Login	102
Gambar 96. Tampilan Antar Muka Menu Pengguna (Administrator)	103
Gambar 97. Tampilan Antar Muka Menu Pengguna (Dosen)	103
Gambar 98. Tampilan Antar Muka Menu Pengguna (Kaprodi)	104
Gambar 99. Tampilan Antar Muka Menu Pengguna (Mahasiswa)	104
Gambar 100. Tampilan Antar Muka Menu Pengguna (Staf)	105
Gambar 101. Tampilan Antar Muka Menu Judul (Administrator)	106
Gambar 102. Tampilan Antar Muka Menu Judul (Dosen)	106
Gambar 103. Tampilan Antar Muka Menu Judul (Kaprodi)	107
Gambar 104. Tampilan Antar Muka Menu Judul (Mahasiswa)	107
Gambar 105. Tampilan Antar Muka Menu Judul (Staf)	108
Gambar 106. Tampilan Antar Muka Form Tambah Data Judul (Administrator)	108
Gambar 107. Tampilan Antar Muka Form Tambah Data Judul (Staf)	109
Gambar 108. Tampilan Antar Muka Detail Data Judul (Administrator)	109
Gambar 109. Tampilan Antar Muka Detail Data Judul (Dosen)	110
Gambar 110. Tampilan Antar Muka Detail Data Judul (Kaprodi)	110
Gambar 111. Tampilan Antar Muka Detail Data Judul (Staf)	110
Gambar 112. Tampilan Antar Muka Menu Upload Proposal (Administrator)	111
Gambar 113. Tampilan Antar Muka Menu Upload Proposal (Mahasiswa)	111
Gambar 114. Tampilan Antar Muka Menu List Proposal (Administrator)	112
Gambar 115. Tampilan Antar Muka Menu List Proposal (Kaprodi)	112
Gambar 116. Tampilan Antar Muka Daftar Dosen (Administrator)	113
Gambar 117. Tampilan Antar Muka Daftar Dosen (Dosen)	113
Gambar 118. Tampilan Antar Muka Daftar Dosen (Kaprodi)	114
Gambar 119. Tampilan Antar Muka Daftar Dosen (Mahasiswa)	114
Gambar 120. Tampilan Antar Muka Daftar Dosen (Staf)	114
Gambar 121. Tampilan Antar Detail Data Dosen (Administrator)	115
Gambar 122. Tampilan Antar Detail Data Dosen (Dosen)	115
Gambar 123. Tampilan Antar Detail Data Dosen (Kaprodi)	116
Gambar 124. Tampilan Antar Detail Data Dosen (Mahasiswa)	116
Gambar 125. Tampilan Antar Detail Data Dosen (Staf)	116
Gambar 126. Tampilan Antar Muka Menu Hak Akses (Administrator)	117

Gambar 127. Tampilan Antar Muka Menu Hak Akses (Staf)	117
Gambar 128. Tampilan Antar Muka Menu Daftar Hak Akses (Administrator)	118
Gambar 129. Tampilan Antar Muka Menu Daftar Hak Akses (Staf)	118
Gambar 130. Tampilan Antar Muka Form Tambah Hak Akses	119
Gambar 131. Tampilan Antar Muka Menu Pesan (Administrator)	119
Gambar 132. Tampilan Antar Muka Menu Pesan (Dosen).....	120
Gambar 133. Tampilan Antar Muka Menu Pesan (Kaprodi).....	120
Gambar 134. Tampilan Antar Muka Menu Pesan (Mahasiswa).....	120
Gambar 135. Tampilan Antar Muka Menu Pesan (Staf)	121
Gambar 136. Tampilan Antar Muka Buat Pesan (Administrator)	121
Gambar 137. Tampilan Antar Muka Buat Pesan (Dosen)	122
Gambar 138. Tampilan Antar Muka Buat Pesan (Kaprodi)	122
Gambar 139. Tampilan Antar Muka Buat Pesan (Mahasiswa).....	122
Gambar 140. Tampilan Antar Muka Buat Pesan (Staf)	123
Gambar 141. Tampilan Antar Muka Baca Pesan (Administrator).....	123
Gambar 142. Tampilan Antar Muka Baca Pesan (Dosen).....	124
Gambar 143. Tampilan Antar Muka Baca Pesan (Kaprodi)	124
Gambar 144. Tampilan Antar Muka Baca Pesan (Mahasiswa)	124
Gambar 145. Tampilan Antar Muka Baca Pesan (Staf).....	125
Gambar 146. Tampilan Antar Muka Menu Users (Administrator).....	125
Gambar 147. Tampilan Antar Muka Menu Users (Staf).....	126
Gambar 148. Tampilan Antar Muka Detail Users (Administrator)	126
Gambar 149. Tampilan Antar Muka Detail Users (Staf)	127
Gambar 150. Tampilan Antar Muka Form Tambah Users (Administrator)	127
Gambar 151. Tampilan Antar Muka Form Tambah Users (Staf)	128

BAB I

PENDAHULUAN

A. Latar Belakang

Saat ini perkembangan teknologi sangat pesat dan peranannya sangat penting untuk mendukung aktivitas manusia agar dapat mengoptimalkan waktu dengan lebih baik. Salah satu contoh teknologi yang membantu aktivitas manusia adalah sistem informasi. Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengelolaan data, mendukung operasi, bersifat manajerial dan strategi kegiatan dari suatu organisasi atau instansi dan menyediakan laporan-laporan bagi pihak tertentu. Instansi yang bergerak di bidang pendidikan seperti Universitas juga membutuhkan dukungan sistem informasi dalam peningkatan mutu pelayanan terhadap proses pendidikan, seperti sistem informasi akademik yang terdiri dari banyak proses antara lain perencanaan perkuliahan hingga proses akhir seperti bimbingan skripsi.

Menurut Anik Ghufroon, dkk (2013: 1), tugas akhir atau skripsi adalah karya tulis ilmiah mahasiswa, yang merupakan kulminasi proses berpikir ilmiah sesuai dengan disiplin ilmunya, yang disusun untuk memenuhi persyaratan memperoleh gelar sesuai dengan jenjangnya. Di Jurusan Pendidikan Matematika (Jurdikmat) Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Universitas Negeri Yogyakarta (UNY), mahasiswa yang ingin mengajukan proposal skripsi harus mencari informasi skripsi yang sudah ada terutama skripsi yang membahas masalah yang serupa dengan masalah yang ingin diangkat oleh mahasiswa yang ingin mengajukan skripsi. Kemudian, mahasiswa yang proposal skripsinya disetujui oleh

Ketua Program Studi, dapat memulai proses bimbingan skripsi. Dalam prakteknya, aktivitas bimbingan tugas akhir di Jurdikmat UNY dilakukan dengan diskusi, tanya jawab, dan pemberian masukan melalui tatap muka antar mahasiswa dan dosen pembimbing.

Dalam proses bimbingan tugas akhir skripsi Jurdikmat UNY, terdapat beberapa kendala yang menyebabkan proses pengerjaan tugas akhir mahasiswa terhambat. Kendala-kendala tersebut antara lain sebagai berikut:

1. Dokumentasi judul skripsi yang sudah digunakan kurang tertata karena hanya ditulis di buku daftar skripsi Jurdikmat UNY, sehingga mahasiswa harus membaca judul skripsi yang sudah digunakan satu persatu agar tidak menggunakan judul yang sama.
2. Kesulitan mahasiswa dan dosen untuk mengadakan pertemuan dalam rangka bimbingan tugas akhir skripsi dikarenakan perbedaan jadwal kesibukan masing-masing individu.

Untuk mengatasi permasalahan tersebut, perlu dirancang sistem pendukung bimbingan skripsi, yang dapat membantu menyelesaikan permasalahan yang dihadapi dalam proses bimbingan skripsi di Jurdikmat UNY saat ini dengan menerapkan teknologi informasi sehingga dapat bekerja layaknya bimbingan skripsi secara tatap muka dengan perantara sistem.

Berdasarkan hal-hal tersebut, muncullah gagasan untuk membuat Aplikasi Bimbingan Tugas Akhir Skripsi *Online* untuk mahasiswa. Aplikasi ini diharapkan akan memberikan kemudahan baik bagi para mahasiswa maupun dosen pembimbing untuk melakukan bimbingan skripsi sehingga dapat meminimalisir

proses pembuatan tugas akhir skripsi akibat perbedaan jadwal kesibukan antara mahasiswa dan dosen pembimbing. Dengan aplikasi ini mahasiswa dan dosen pembimbing diharapkan tetap dapat melakukan bimbingan tanpa harus mencocokkan jadwal atau bertatap muka.

B. Batasan Masalah

1. Aplikasi Bimbingan Tugas Akhir Skripsi *Online* ini dibuat dalam ruang lingkup Jurdikmat UNY.
2. Aktivitas terkait tugas akhir skripsi yang tidak dapat dilaksanakan secara *online* tidak dibahas dan diasumsikan sudah dilaksanakan secara manual (*offline*).
3. Aplikasi dirancang dengan bahasa pemrograman PHP dengan memanfaatkan *framework* Yii, dan *Database Management System* (DBMS) MySQL.
4. Data informasi terkait tugas akhir dalam Aplikasi Bimbingan Tugas Akhir Skripsi *Online* menggunakan data dari Jurdikmat UNY.

C. Rumusan Masalah

Berdasarkan latar belakang yang ada, masalah yang dapat dirumuskan dalam tugas akhir ini adalah:

1. Bagaimana rancangan Aplikasi Bimbingan Tugas Akhir Skripsi *Online*?
2. Apakah program yang dirancang dapat membantu proses bimbingan tugas akhir Jurdikmat UNY?

D. Tujuan Penelitian

1. Merancang sistem yang dapat membantu proses bimbingan tugas akhir mahasiswa Juridimat FMIPA UNY.
2. Merancang aplikasi *online* menggunakan bahasa pemrograman PHP dengan memanfaatkan *framework* Yii, dan *Database Management System* (DBMS) MySQL.

E. Manfaat Penelitian

Manfaat dari penelitian ini adalah menghasilkan suatu aplikasi yang dapat membantu proses bimbingan tugas akhir skripsi di Juridimat FMIPA UNY.

BAB II

KAJIAN TEORI

A. Deskripsi Teori

Beberapa teori yang digunakan sebagai dasar dalam penelitian ini adalah (1) pengertian tugas akhir di UNY, (2) pengertian teknologi informasi, (3) definisi basis data, (4) Model Basis Data Relasional, (5) definisi *Database Management Sistem* (DBMS) sebagai sistem yang mengelola *database*, (6) metode perancangan sistem, (7) *Flowmap*, (8) *Data Flow Diagram*, (9) PHP: *Hypertext Preprocessor* (PHP) sebagai bahasa skrip dalam program, (10) MySQL sebagai perangkat lunak DBMS, (11) *framework* Yii sebagai kerangka kerja, dan (12) *Software Requirements Specification* (SRS) sebagai dokumen yang dibuat oleh pengembang *software* yang menjelaskan tentang berbagai kebutuhan yang harus dipenuhi oleh suatu *software*, (13) Jaminan Kualitas Perangkat Lunak, dan (14) Pengujian *software*. Penjelasan lebih lanjut mengenai teori-teori tersebut adalah sebagai berikut:

1. Tugas Akhir di UNY

Mahasiswa perlu melakukan konsultasi atau bimbingan dalam pengerjaan tugas akhir kepada dosen pembimbing yang telah ditunjuk oleh ketua program studi Jurusan Pendidikan Matematika UNY agar dapat membantu mahasiswa dalam mengerjakan Tugas Akhir menjadi lebih terarah dan lebih fokus sehingga hasil yang diharapkan dapat tercapai.

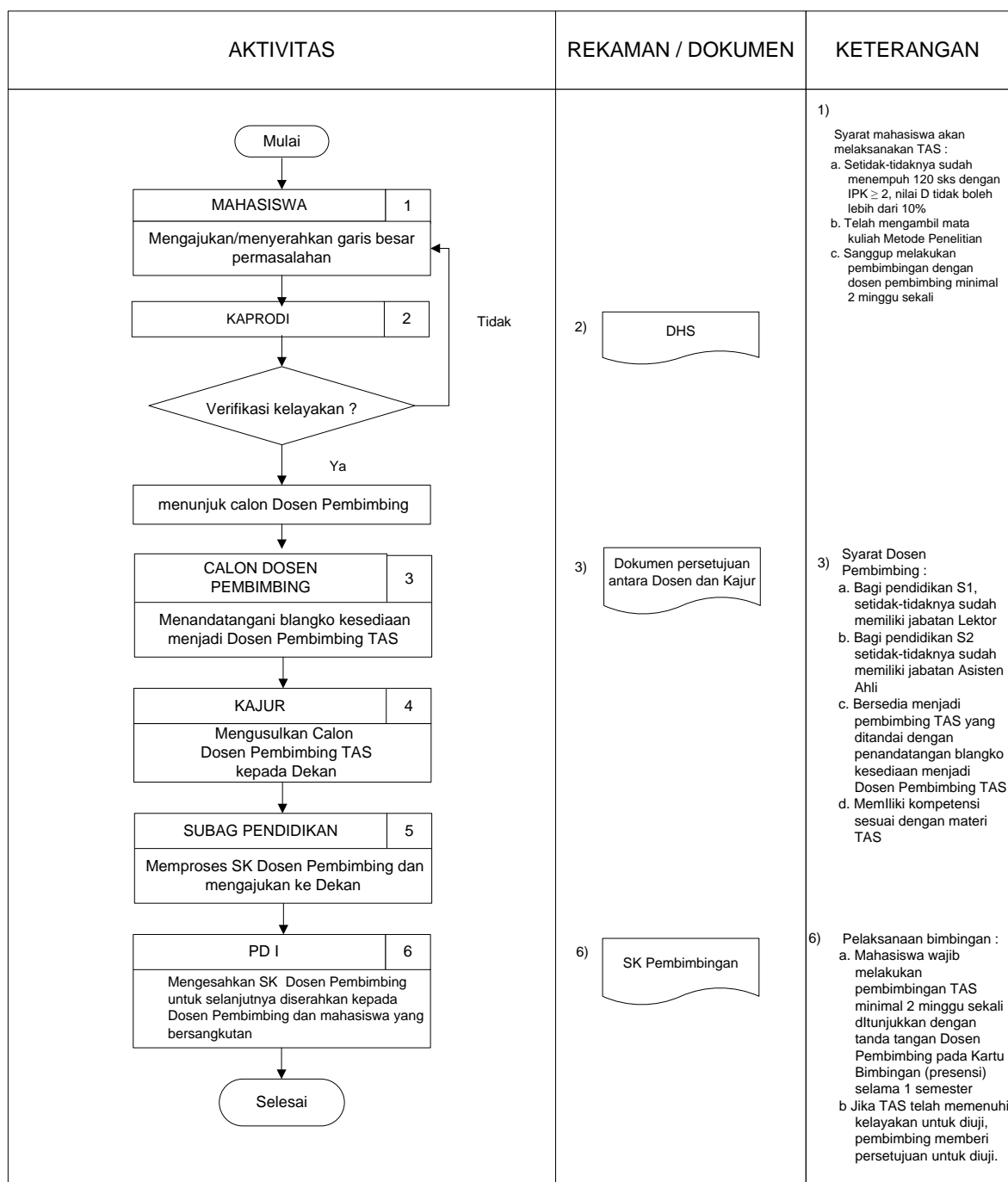
Penyusunan tugas akhir bertujuan memantapkan wawasan pengetahuan dan pengalaman mahasiswa dalam melakukan kegiatan ilmiah, dengan cara memberi kesempatan memformulasikan ide, konsep, pola berpikir, dan kreativitas yang

dikemas secara komprehensif, dan mengomunikasikan dalam format yang lazim digunakan di kalangan masyarakat ilmiah (Anik Ghufro, dkk, 2013: 2).

Di Jurdikmat UNY terdiri dari beberapa tahap, yaitu:

- a. Pengajuan proposal oleh mahasiswa kepada Ketua Program Studi.
- b. Penunjukkan dosen pembimbing oleh Ketua Program Studi.
- c. Proses bimbingan skripsi mahasiswa dan dosen pembimbing.
- d. Pendaftaran ujian oleh mahasiswa.

Detail dari tahap-tahap tersebut disajikan pada Gambar 1 dan Gambar 2.

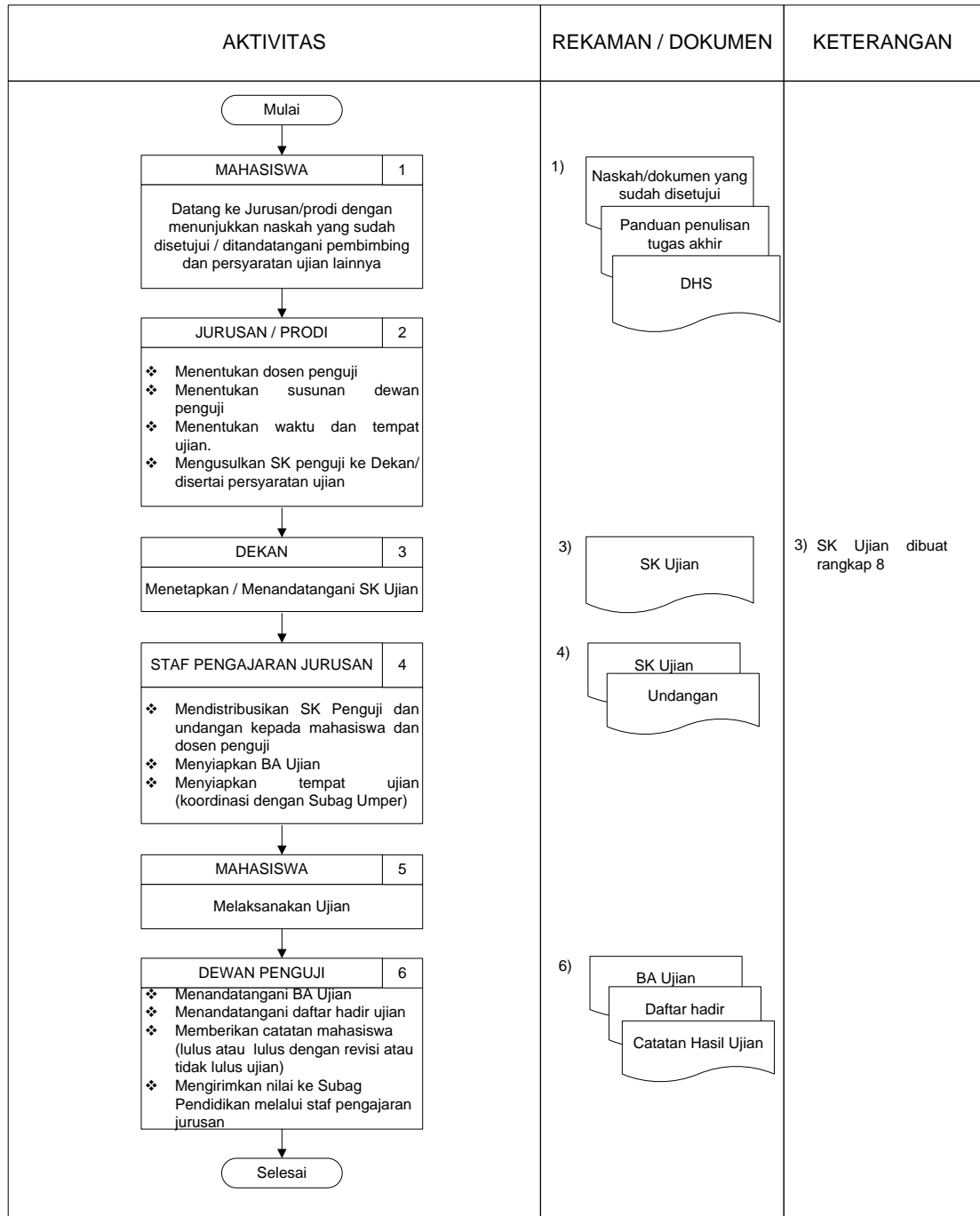


Gambar 1. Alur Pendaftaran Skripsi

Keterangan:

Langkah 2 juga membutuhkan dokumen proposal skripsi.

Langkah 4 dan 5 membutuhkan dokumen melalui subag pendidikan.



Gambar 2. Alur Pendaftaran Ujian Skripsi

2. Sistem Informasi

Menurut Kenneth C Laudon dan Jane P Laudon (2008:15), Sistem informasi adalah sekumpulan komponen yang saling berhubungan, mengumpulkan (atau mendapatkan), memproses, menyimpan, dan mendistribusikan informasi untuk

menunjang pengambilan keputusan dan pengawasan dalam suatu organisasi. Menurut Aprillita Dwiyani (2013: 2), sistem informasi adalah sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, membantu dan mendukung kegiatan operasi, bersifat manajerial dari suatu organisasi dan membantu mempermudah penyediaan laporan yang diperlukan.

Al Fatta (2007: 10) mengemukakan bahwa komponen-komponen yang menyusun suatu sistem informasi terdiri dari:

- a. Perangkat keras, yaitu perangkat keras komponen untuk melengkapi kegiatan memasukan data, memroses data, dan keluaran data.
- b. Perangkat lunak, yaitu program dan instruksi yang diberikan ke komputer.
- c. Database, yaitu kumpulan data dan informasi yang diorganisasikan sedemikaian rupa sehingga mudah diakses pengguna sistem informasi.
- d. Telekomunikasi, yaitu komunikasi yang mnghubungkan antara pengguna sistem dengan sistem komputer secara bersama-sama ke dalam suatu jaringan kerja yang efektif.
- e. Manusia, yaitu personel dari sistem informasi, meliputi admin, analis, programer dan operator.

Al Fatta (2007: 12) mengemukakan bahwa sistem informasi dapat dibagi menjadi beberapa tipe yaitu:

- a. *Transaction Proccesing System* (TPS)

Transaction Proccesing System atau Sistem Pemrosesan Transaksi adalah sistem informasi terkomputerisasi yang dikembangkan untuk memroses sejumlah data untuk transaksi bisnis rutin.

b. Sistem Informasi Manajemen (SIM)

Sistem Informasi Manajemen adalah sebuah sistem informasi yang berfungsi untuk membantu perencanaan, pengendalian, dan pengambilan keputusan dengan menyediakan resume rutin dan laporan-laporan tertentu.

c. *Decision Support System* (DSS)

Decision Support System atau Sistem Penunjang Keputusan adalah sistem informasi yang berfungsi mengombinasikan data dan model analisis canggih atau peralatan data analisis untuk mendukung pengambilan keputusan yang semiterstruktur maupaun tidak terstruktur.

d. *Expert System* (ES)

Expert System atau Sistem Pakar adalah representasi pengetahuan yang menggambarkan cara seorang ahli dalam mendekati suatu masalah.

3. Basis Data

Basis data terdiri dari dua kata yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang tempat bersarang dan berkumpul, sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, symbol, teks, gambar, bunyi, atau kombinasinya (Syahril Akbar, 2010: 43).

Menurut Firman Kemal P., (2010: 22-23), basis data dapat didefinisikan dalam sejumlah sudut pandang, diantaranya:

- a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu untuk memenuhi berbagai kebutuhan.
- c. Kumpulan file/tipe/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

4. Model Basis Data Relasional

Abdul Kadir (2003: 270-276) mengemukakan bahwa model basis data adalah sekumpulan konsep terintegrasi yang dipakai untuk menjabarkan data, hubungan antar data, dan kekangan terhadap data yang digunakan untuk menjaga konsistensi. Model data terbagi menjadi 4 macam, yaitu:

- a. Model data hirarkis.
- b. Model data jaringan.
- c. Model data relasional.
- d. Model data berbasis objek.

Model data relasional merupakan model yang paling sederhana sehingga mudah digunakan dan dipahami oleh pengguna. Model ini menggunakan sekumpulan tabel dengan masing-masing tabel tersusun atas sejumlah baris dan kolom. Pada model data relasional, kaitan antara 2 buah tabel disebut relasi. Hal-hal yang perlu diperhatikan dalam menyusun tabel-tabel dalam basis data relasional adalah sebagai berikut:

a. Atribut Kunci

Penentuan kunci merupakan hal yang paling penting dalam basis data relasional. Kunci sebagai metode untuk mengakses suatu baris tertentu sekaligus menjadi pengenalan unik (tidak ada yang sama) rekaman di dalam suatu tabel. Kunci dapat berupa sebuah atribut atau gabungan dari beberapa atribut.

Menurut Andri Kristanto (2003: 76-77), atribut kunci bagi menjadi:

1) Kunci kandidat (*Candidat key*)

Kunci kandidat adalah suatu atribut yang mengidentifikasi secara unik dari suatu kejadian yang sifatnya khusus dari suatu entiti. Kunci kandidat ini tidak ganda.

2) Kunci primer (*Primary key*)

Kunci primer adalah kunci kandidat yang dipilih untuk mewakili setiap kejadian dari suatu entiti.

3) Kunci alternatif (*Alternative key*)

Kunci alternatif adalah kunci kandidat yang tidak dipilih sebagai kunci primer. Kunci alternatif lebih diorientasikan untuk pengurutan rekaman di dalam sebuah tabel atau entiti.

4) Kunci tamu (*Foreign key*)

Kunci tamu adalah kunci primer suatu entiti yang dihubungkan pada entiti lain. Biasanya jika ada dua tabel, kunci primer akan diletakkan pada entiti induk sedangkan pada entiti anak diletakkan kunci tamu.

b. Normalisasi

Menurut Kronke (1992) dalam Abdul Kadir (1999: 65) mengemukakan bahwa normalisasi sebagai proses untuk mengubah suatu tabel yang memiliki masalah tertentu ke dalam 2 tabel atau lebih yang tak memiliki masalah tersebut. Tujuan dari normalisasi adalah untuk mengurangi redudansi dan ketidakkonsistenan data. Aturan-aturan normalisasi dinyatakan dalam istilah bentuk normal. Bentuk normal adalah suatu aturan yang dikenakan pada tabel-tabel dalam basis data. Suatu tabel dikatakan berada dalam bentuk normal, jika memenuhi aturan-aturan pada level normalisasi yang bersesuaian. Menurut Abdul Kadir (2001: 54) aturan-aturan dalam masing-masing normalisasi tersebut adalah sebagai berikut:

1) Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan disimpan, tidak ada keharusan mengikuti suatu format tertentu, mungkin saja data tidak lengkap atau data terduplikasi dan data dikumpulkan apa adanya.

2) Bentuk normal pertama

Suatu tabel dikatakan dalam bentuk normal pertama (1NF) bila setiap kolom bernilai tunggal untuk setiap baris. Ini berarti bahwa nama kolom yang berulang cukup diwakili oleh sebuah nama kolom.

3) Bentuk normal kedua

Suatu tabel berada dalam bentuk normal kedua (2NF) jika tabel berada dalam bentuk normal pertama, semua kolom bukan kunci

primer tergantung sepenuhnya terhadap kunci primer. Suatu kolom disebut tergantung sepenuhnya terhadap kunci primer jika nilai pada suatu kolom selalu bernilai sama untuk suatu kunci primer yang sama.

4) Bentuk normal ketiga

Suatu tabel berada dalam bentuk normal ketiga (3NF) jika tabel berada dalam bentuk normal kedua, setiap kolom yang bukan kunci primer tidak memiliki ketergantungan secara transitif terhadap kunci primer.

5. Database Management Sistem (DBMS)

Menurut Syahril Akbar (2010: 46), pengelolaan basis data secara spesifik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak (sistem) yang khusus/spesifik. Perangkat inilah yang disebut DBMS yang menentukan bagaimana data diorganisasi, disimpan, diubah, dan diambil kembali. DBMS menerapkan mekanisme pengamanan data secara bersama, pemaksaan keakuratan/ konsistensi data dan sebagainya.

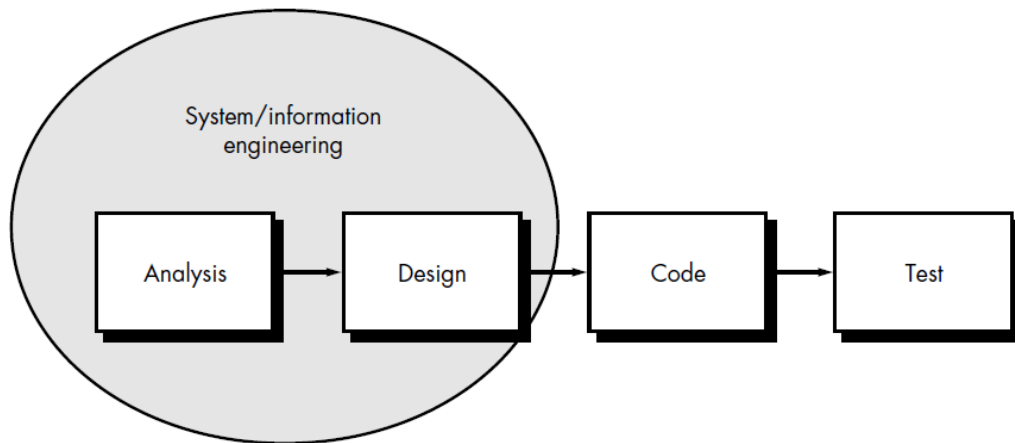
DBMS berfungsi menyediakan sebuah cara untuk menyimpan dan menerima kembali informasi database secara mudah dan efisien. Raharjo (2011: 10) memaparkan macam-macam program DBMS yang saat ini sering digunakan adalah MySQL, Oracle, Interbase/Firebird, IBM DB2.

6. Model Perancangan Sistem

Perancangan sistem merupakan suatu gambaran secara umum kepada pengguna/user tentang sistem yang akan dibuat. Desain sistem akan dibuat untuk

mengambarkan komponen-komponen sistem informasi yang akan didesain secara rinci. Model perancangan sistem yang akan dipakai menggunakan metode analisis sistem terstruktur atau lebih dikenal dengan model *Waterfall*.

Model *Waterfall* sering disebut dengan “*Classic Life Cycle*”. Ini merupakan model yang muncul pertama kali, sekitar tahun 1970 sehingga sering dianggap kuno tetapi merupakan model yang paling banyak dipakai didalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap *Analysis, Desain, Code, Test*.



Gambar 3. Model *Waterfall* Presmann

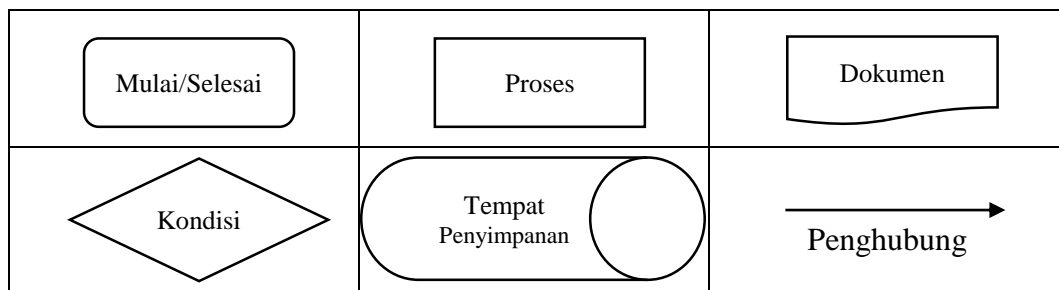
Gambar 3 merupakan tahapan secara umum dari model *Waterfall*. Menurut Pressman (2001: 28) model *Waterfall* diuraikan dengan tahap-tahap sebagai berikut:

- a. *Analysis* adalah tahap menganalisa hal-hal yang diperlukan dalam pelaksanaan perancangan sistem.
- b. *Design* adalah tahap penerjemah atau tahap perancangan dari keperluan-keperluan yang dianalisis dalam bentuk yang lebih mudah dimengerti oleh pemakai.

- c. *Code* adalah tahap implementasi dari hasil sistem yang telah dirancang dalam bahasa pemrograman yang telah ditentukan dan digunakan dalam pembuatan sistem.
- d. *Test* adalah tahap pengujian terhadap program yang telah dibuat. Pengujian dilakukan agar fungsi-fungsi dalam sistem bebas dari error, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

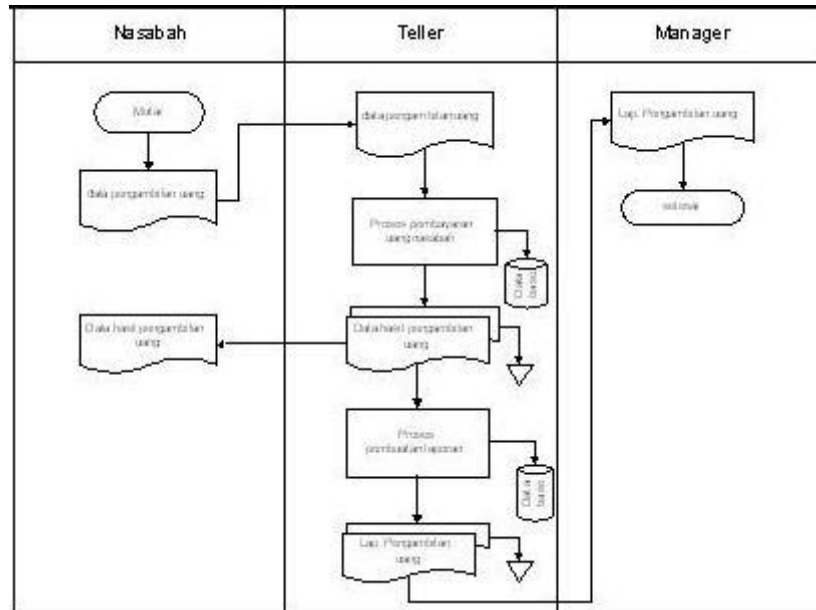
7. *Flowmap*

Menurut Firman (2010: 25), *flowmap* adalah gambaran hubungan antara entitas yang terlihat berupa alihan-alihan dokumen yang ada. Bagan alir dokumen merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusannya. Keterangan gambar dalam flowmap disajikan pada Gambar 4.



Gambar 4. Keterangan Gambar Dalam Flowmap

Contoh flowmap disajikan pada Gambar 5.



Gambar 5. Contoh Flowmap

8. Data Flow Diagram (DFD)

Firman (2010: 26) menyampaikan bahwa DFD memperlihatkan fungsional dari nilai yang dihitung oleh sistem, termasuk nilai masukan, nilai keluaran, serta tempat penyimpanan internal. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur (*structured analysis & design*). DFD dapat menggambarkan arus data di dalam sistem dengan terstruktur. DFD memberikan informasi tambahan yang digunakan selama analisis domain informasi dan berfungsi sebagai dasar bagi pemodelan fungsi.

Al-Bahra bin Ladjamuddin (2005: 67-75), elemen dasar dari DFD adalah sebagai berikut:

a. Kesatuan luar (*External entity*)

Kesatuan luar adalah sesuatu yang berada di luar sistem tetapi memberikan data ke dalam sistem atau sebaliknya. Kesatuan luar tidak termasuk bagian dari sistem. Pedoman pemberian nama kesatuan luar adalah sebagai berikut:

- 1) Nama kesatuan luar berupa kata benda.
- 2) Kesatuan luar tidak boleh memiliki nama yang sama kecuali memang objeknya sama.

b. Proses (*Process*)

Proses merupakan kegiatan atau kerja yang dilakukan oleh sistem. Proses berfungsi mentransformasikan satu atau beberapa data masukan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan.

Pedoman pemberian nama proses adalah sebagai berikut:

- 1) Nama proses terdiri dari kata kerja dan kata benda yang mencerminkan fungsi proses tersebut.
- 2) Kata proses tidak boleh digunakan sebagai bagian dari nama suatu proses.
- 3) Tidak boleh ada beberapa proses yang memiliki nama yang sama.
- 4) Proses harus diberi nomor.

c. Simpanan data (*Data store*)

Simpanan data merupakan tempat penyimpanan data yang ada dalam sistem. Pedoman pemberian nama simpanan data adalah sebagai berikut:

- 1) Nama harus mencerminkan simpanan data tersebut.
- 2) Bila namanya lebih dari satu kata, maka harus diberi tanda sambung.

d. Arus data (*Data flow*)

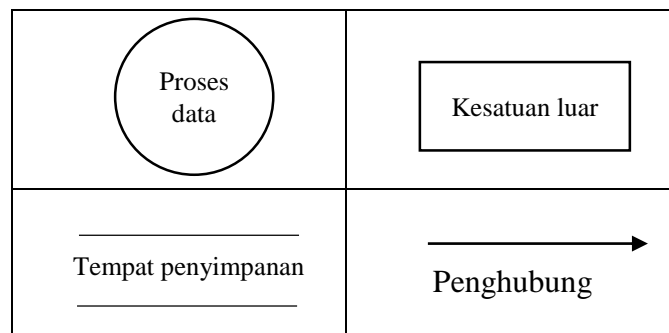
Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data

ditunjukkan dengan arah panah dan garis diberi nama atas arus data yang mengalir. Pedoman pemberian nama aliran data adalah sebagai berikut:

- 1) Nama aliran data yang terdiri dari beberapa kata dihubungkan dengan garis sambung.
- 2) Sedapat mungkin nama aliran data ditulis lengkap.
- 3) Tidak boleh ada aliran data dari kesatuan luar ke simpanan data atau sebaliknya, hubungan kesatuan luar dengan simpanan data harus melalui proses.

e. Simbol DFD

Simbol DFD disajikan pada Gambar 6.



Gambar 6. Keterangan Gambar Dalam DFD

Menurut Agus Winarto (2007: 2) DFD dibagi menjadi 2 jenis, yaitu:

a. Diagram Konteks

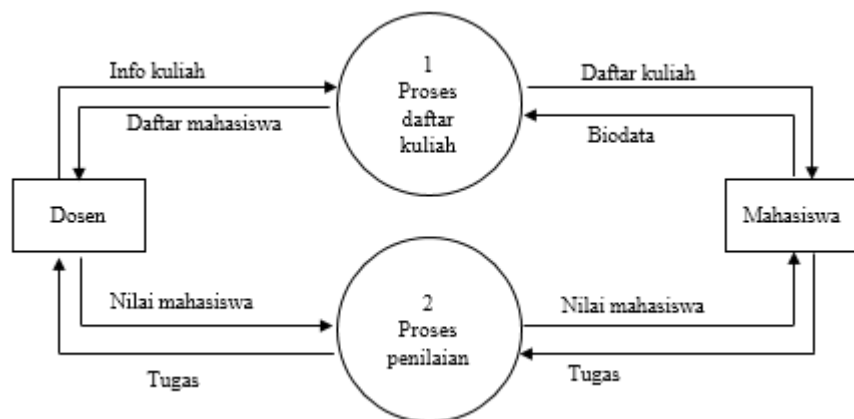
Diagram konteks merupakan DFD Level 0, yaitu diagram yang paling sederhana dari sebuah sistem informasi yang menggambarkan aliran data dari kesatuan luar ke dalam sistem dan sebaliknya. Contoh Diagram Konteks disajikan pada Gambar 7.



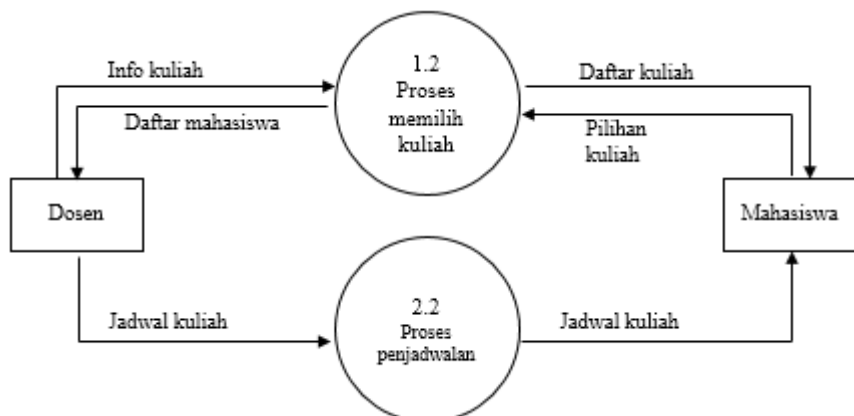
Gambar 7. Contoh Diagram Konteks

b. DFD Level n

Diagram Level n merupakan diagram yang digunakan untuk menggambarkan diagram hasil pengembangan dari Diagram Konteks ke dalam komponen yang lebih detail. Contoh DFD Level 1 disajikan pada Gambar 8, dan contoh DFD Level 2 disajikan pada Gambar 9.



Gambar 8. Contoh DFD Level 1



Gambar 9. Contoh DFD Level 2

9. PHP

Menurut Badiyanto (2013: 32), PHP: *Hypertext Preprocessor* (PHP) adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML/PHP banyak digunakan untuk membuat situs web dinamis.

Tujuan utama penggunaan bahasa ini adalah untuk memungkinkan perancang dan menulis halaman web menjadi dinamis dengan cepat. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995, seorang programmer C. Pada waktu itu, PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*.

Menurut Deni Sutaji (2012: 2), aturan penulisan skrip PHP adalah sebagai berikut:

- a. Embedded script, yaitu dengan cara meletakkan tag PHP diantara tag-tag

HTML. Contohnya:

```
<html>
<body>
<?php echo "Belajar"; ?>
</body>
</html>
```

- b. Non embedded script, yaitu dengan cara semua skrip HTML diletakkan dalam skrip PHP. Contohnya:

```
<?php
echo "<html>";
echo "<body>";
```

```
echo "Belajar PHP";
```

```
echo "</body>";
```

```
echo "</html>";
```

Deni Sutaji (2012: 3) menjelaskan bahwa PHP memiliki variabel yang digunakan sebagai tempat penyimpanan data sementara. Data tersebut akan hilang setelah program selesai dieksekusi. Aturan penggunaan nama variabel adalah sebagai berikut:

- a. Diawali dengan karakter \$.
- b. Bersifat *case sensitive*, jadi \$Var berbeda dengan \$var atau \$VAR.
- c. Karakter pertama harus berupa huruf atau garis bawah (_).
- d. Karakter berikutnya boleh huruf, angka atau garis bawah (_).

Berikut adalah beberapa contoh penulisan dan pendeklarasian variabel:

```
$namadepan = "Ali Shodikin";
```

```
$harga_barang = 3500;
```

```
$nilai3 = 3.45;
```

10. MySQL

Menurut Aprillita (2013: 2), MySQL adalah salah satu perangkat lunak Database Management System (DBMS) yang sering digunakan saat ini, yang didistribusikan secara gratis di bawah lisensi GPL (GNU General Public License). Sehingga setiap orang mudah untuk mendapatkan dan bebas untuk menggunakan MySQL.

Menurut Deni Sutaji (2011: 44), SQL dibagi menjadi tiga bentuk query, yaitu :

a. DDL (*Data Definition Language*)

DDL adalah sebuah Metode Query SQL yang berguna untuk mendefinisikan data pada sebuah database. Berikut *query* yang termasuk DDL:

- 1) CREATE digunakan untuk melakukan pembuatan tabel dan database.
- 2) DROP digunakan untuk melakukan penghapusan tabel maupun database.
- 3) ALTER digunakan untuk melakukan perubahan struktur tabel yang telah dibuat, baik menambah Field (add), mengganti nama Field (change) ataupun menamakannya kembali (rename), serta menghapus (drop).

b. DML (*Data Manipulation Language*)

DML adalah sebuah metode *query* yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari *query* ini adalah untuk melakukan pemanipulasian database yang telah ada atau yang telah dibuat sebelumnya. Berikut *query* yang termasuk DML:

- 1) SELECT digunakan untuk menampilkan data pada tabel.
- 2) INSERT digunakan untuk melakukan penginputan / pemasukan data pada tabel database.
- 3) UPDATE digunakan untuk melakukan perubahan atau peremajaan terhadap data yang ada pada tabel.
- 4) DELETE digunakan untuk melakukan penghapusan data pada tabel.

c. DCL (*Data Control Language*)

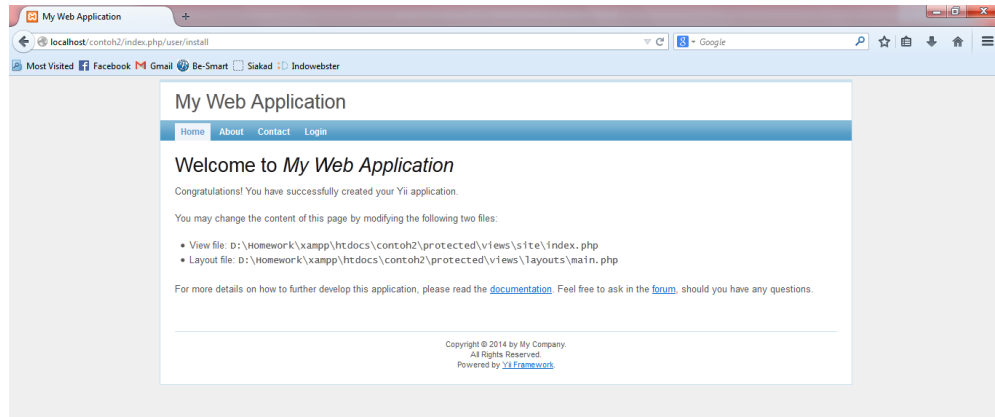
DCL adalah sebuah metode Query yang dapat digunakan untuk mengendalikan eksekusi perintah. Biasanya berhubungan dengan pengaturan hak akses. Berikut *query* yang termasuk DCL:

- 1) GRANT : Digunakan untuk memberikan hak akses (*privilage*) kepada user tertentu.
- 2) REVOKE : Digunakan untuk mencabut hak akses dari user tertentu

11. *Framework Yii*

Johan I, dkk (2011: 7) memaparkan bahwa *framework* adalah desain yang bisa digunakan kembali dan biasanya dinyatakan sebagai satu set abstraksi *class* yang mengatur bagaimana *class* saling terhubung. Perancangan pada *framework* dibuat sedemikian rupa sehingga sebagian atau seluruh software dapat digunakan kembali. *Framework* merupakan kerangka kode yang dapat disimpulkan dengan *classes* yang spesifik atau dengan fungsi yang telah dirancang untuk mengatasi masalah yang dihadapi.

Framework Yii adalah kerangka kerja PHP berbasis komponen yang dibuat oleh Qiang Xue tanggal 1 Januari 2008 untuk pengembangan aplikasi *web* berskala besar. *Framework Yii* menyediakan *reusablility* maksimum dalam pemrograman *web* dan mampu meningkatkan kecepatan dalam membuat aplikasi *web*. Nama *Yii* diambil dari singkatan “Yes It Is!” (Badiyanto, 2013: 7). Aplikasi *web* yang dibangun menggunakan *Yii* menggunakan arsitektur *Model-View-Controller* (MVC). Contoh aplikasi *Yii* disajikan pada Gambar 10.



Gambar 10. Aplikasi Yii

Model-View-Controller (MVC) adalah model pembuatan program yang menerapkan arsitektur yang memisahkan proses, tampilan, dan bagian yang menghubungkan proses dengan tampilan. MVC bertujuan untuk memisahkan proses bisnis dari pertimbangan antarmuka *user* agar para pengembang bisa lebih mudah setiap bagian tanpa memengaruhi yang lain (Badiyanto, 2013: 49). Dalam MVC, *model* menggambarkan informasi (data) dan proses bisnis, *view* (tampilan) berisi elemen antarmuka *user* seperti teks, formulir masukan, sementara *controller* mengatur komunikasi antara *model* dengan *view*. Badiyanto, (2013: 49-55) menjelaskan tentang *Model-View-Controller* sebagai berikut:

a. *Model*

Model merupakan kelas yang mendasari logika proses dalam aplikasi perangkat lunak dan kelas yang terkait dengannya. *Model* adalah suatu objek yang tidak mengandung informasi tentang *user interface*. *Model* juga merupakan suatu kelas yang berisi metode/fungsi dan digunakan untuk menyimpan data dan aturan bisnis yang relevan. Pengguna dapat membuat

model berdasarkan tabel menggunakan *Model Generator*. Contoh penggunaan *Model Generator* disajikan pada Gambar 11.

Generators

- [Controller Generator](#)
- [Crud Generator](#)
- [Form Generator](#)
- [Model Generator](#)
- [Module Generator](#)

Model Generator

This generator generates a model class for the specified database table.

Fields with * are required. Click on the highlighted fields to edit them.

Database Connection *
db

Table Prefix
[empty]

Table Name *
authassignment

Model Class *
Authassignment

Base Class *
CActiveRecord

Model Path *
application.models

Build Relations
☒

Use Column Comments as Attribute Labels
☐


Code Template *
default (D:\Homework\xampp\htdocs\yii\framework\yii\generators\model\templates\default)

Code File	Generate
models\Authassignment.php	new <input checked="" type="checkbox"/>

Gambar 11. Model Generator Yii

b. *View*

View merupakan kumpulan dari kelas yang mewakili unsur-unsur dalam antarmuka *user*. Dalam *view*, terdapat nama yang dipakai untuk mengidentifikasi *file* skrip tampilan saat dipanggil lewat fungsi *render*. Nama *view* sama seperti nama *file* skrip *view*-nya. Sebagai contoh, *file view edit* merujuk pada *file* skrip bernama `edit.php`. Pengguna dapat membuat *file view* yang terdiri dari *create*, *read*, *update*, dan *delete* berdasarkan *model* menggunakan *Crud Generator*. Contoh penggunaan *Crud Generator* disajikan pada Gambar 12.


[help](#) | [webapp](#) | [yii](#) | [logout](#)

Generators

- [Controller Generator](#)
- [Crud Generator](#)
- [Form Generator](#)
- [Model Generator](#)
- [Module Generator](#)

Crud Generator

This generator generates a controller and views that implement CRUD operations for the specified data model.

Fields with * are required. Click on the [highlighted fields](#) to edit them.

Model Class *
Mahasiswa

Controller ID *
mahasiswa

Base Controller Class *
Controller

Code Template *
default (D:\Homework\xampp\htdocs\yii\framework\yii\generators\crud\templates\default)

Code File	Generate
controllers\MahasiswaController.php (diff)	overwrite <input type="checkbox"/>
views\mahasiswa\form.php (diff)	overwrite <input type="checkbox"/>
views\mahasiswa\search.php (diff)	overwrite <input type="checkbox"/>
views\mahasiswa\view.php (diff)	overwrite <input type="checkbox"/>
views\mahasiswa\admin.php (diff)	overwrite <input type="checkbox"/>
views\mahasiswa\create.php (diff)	overwrite <input type="checkbox"/>
views\mahasiswa\index.php (diff)	overwrite <input type="checkbox"/>
views\mahasiswa\update.php (diff)	overwrite <input type="checkbox"/>
views\mahasiswa\view.php (diff)	overwrite <input type="checkbox"/>

Gambar 12. Crud Generator Yii

c. Controller

Controller merupakan kelas yang menghubungkan *model* dan *view*, dan digunakan untuk berkomunikasi antara kelas dalam *model* dan *view*. *Controller* mempunyai *action* standar. Ketika permintaan *user* tidak menetapkan *action* mana yang dijalankan, program akan menjalankan *action* standar. Pengguna dapat membuat *controller* berdasarkan *model* menggunakan *Controller Generator*. Contoh penggunaan *Controller Generator* disajikan pada Gambar 13.

Generators

- [Controller Generator](#)
- [Crud Generator](#)
- [Form Generator](#)
- [Model Generator](#)
- [Module Generator](#)

Controller Generator

This generator helps you to quickly generate a new controller class, one or several controller actions and their corresponding views.

Fields with * are required. Click on the highlighted fields to edit them.

Controller ID *

Base Class *

Action IDs

Code Template *

Code File	Generate
controllers\MahasiswaController.php (diff)	<input type="checkbox"/> overwrite
views\mahasiswa\index.php (diff)	<input type="checkbox"/> overwrite

Gambar 13. *Controller Generator* Yii

d. Alur kerja MVC

Alur kerja yang dilakukan dalam MVC saat membuka halaman *web* adalah sebagai berikut:

- 1) *User* melakukan permintaan dengan memanggil URL misalnya `http://localhost/test/index.php?r=site/login` (test adalah nama aplikasi yang dibuat), dan *web server* memberikan responsi permintaan dengan menjalankan skrip *bootstrap* `index.php`.
- 2) Skrip *bootstrap* membuat sebuah instan aplikasi dan menjalankannya.
- 3) Aplikasi mendapat rincian informasi permintaan *user* dari komponen aplikasi bernama *request*.
- 4) Aplikasi menentukan *controller* dan aksi yang terdapat pada fungsi *action* yang diminta dengan bantuan komponen aplikasi bernama *urlManager*. Dalam contoh ini, *controller* adalah `Site` yang merujuk

pada kelas *SiteController*, dan aksi adalah *login.php* yang arti sebenarnya ditentukan oleh *controller*.

- 5) Aplikasi membuat *instance controller* yang diminta untuk menangani permintaan *user*. *Controller* menentukan aksi *login.php* merujuk pada sebuah metode bernama *actionLogin* dalam kelas *controller*. Kemudian membuat dan menjalankan *filter* terkait aksi ini. Aksi dijalankan jika diijinkan oleh *filter*.
- 6) Aksi membaca *model* *LoginForm* dengan parameter *\$model* yang terdapat pada *controller* dalam *actionLogin* diambil dari pernyataan:

```
$model->attributes=$_POST['LoginForm'];
```
- 7) Aksi menyiapkan *view* (tampilan) bernama *login.php* dengan *model* *LoginForm*.
- 8) *View* membaca dan menampilkan atribut *model* *login.php*.
- 9) *View* menjalankan beberapa *widget* dan *view* menyiapkan hasil yang dipasangkan dalam *layout* (tata letak).

12. *Software Requirements Specification (SRS)*

Secara sederhana, *Software Requirement Specifications (SRS)* adalah dokumen yang menjelaskan tentang berbagai kebutuhan yang harus dipenuhi oleh suatu *software*. Dokumen ini dibuat oleh *developer* (pengembang *software*) setelah menggali informasi dari calon pemakai *software*. Pembuatannya mengikuti standar yang ada dan paling dianggap benar oleh para praktisi rekayasa *software* di dunia. Oleh karena itu, standar yang akan dibahas di sini adalah standar dari IEEE, singkatan dari *Institute of Electrical and Electronics Engineers* (Wisnu, 2012).

SRS yang baik akan bermanfaat bagi *customer*, *supplier*, atau perorangan.

Manfaat-manfaat tersebut antara lain sebagai berikut:

- a. Sebagai bentuk perjanjian antara *customer* dan *supplier* tentang *software* apa yang akan dibuat.
- b. Mengurangi beban dalam proses pengembangan *software*.
- c. Sebagai bahan perkiraan biaya dan rencana penjadwalan.
- d. Sebagai dasar validasi dan verifikasi *software* di ujung penyelesaian proyek nantinya.
- e. Memfasilitasi *transfer*, semisal *software* tersebut ingin di-*transfer* ke pengguna atau mesin-mesin yang lain. *Customer* akan lebih mudah jika ingin men-*transfer software* ke bagian-bagian lain dalam organisasinya. Bahkan, jika terjadi pergantian personil *developer*, proyek dapat mudah di-*transfer* ke personil baru dengan memahami SRS ini.
- f. Mendasari perbaikan produk *software* di kemudian hari. Jadi, SRS boleh diperbaiki dengan alasan dan mekanisme tertentu serta atas kesepakatan antara *customer* dan *developer*.

Dengan SRS, pengguna dapat mencurahkan semua keinginannya terkait *software* dengan jelas dan akurat sehingga *developer* dapat memahami apa yang diinginkan pengguna dengan tepat. Standar ini dapat membantu dalam mengembangkan *outline* SRS yang baku untuk perusahaan pribadi, membantu membuat dokumen SRS dengan format dan isi yang standar (minimal), serta membantu mengembangkan rincian-rincian pendukung lainnya.

13. Jaminan Kualitas Perangkat Lunak

Manurut Pressman dalam Nastiti (2012: 35), jaminan kualitas perangkat lunak adalah aktivitas pelindung yang diaplikasikan pada seluruh proses perangkat lunak. Tujuan dari jaminan kualitas adalah untuk memberikan data yang diperlukan oleh manajemen dan menginformasikan masalah kualitas produk, sehingga dapat memberikan kepastian dan konfidensi bahwa kualitas produk dapat memenuhi sasaran, tidak hanya berkualitas menurut pengembang tapi juga berkualitas dan sesuai dengan keinginan pengguna.

Menurut McCall dalam Nastiti (2012: 36), faktor-faktor penentu kualitas perangkat lunak adalah sebagai berikut:

- a. *Correctness*, sejauh mana suatu perangkat lunak memenuhi spesifikasi dan tujuan penggunaan perangkat lunak dari user.
- b. *Reliability*, sejauh mana keakuratan suatu perangkat lunak dalam melaksanakan fungsinya.
- c. *Efficiency*, banyaknya sumber daya komputasi dan kode program yang dibutuhkan suatu perangkat lunak untuk melakukan fungsinya.
- d. *Integrity*, sejauh mana akses ke perangkat lunak dan data oleh pihak yang tidak berhak dapat dikendalikan; dan
- e. *Usability*, usaha yang diperlukan untuk mempelajari, mengoperasikan, menyiapkan input, dan mengartikan output dari perangkat lunak.
- f. *Maintainability*, usaha yang diperlukan untuk menetapkan dan memperbaiki kesalahan dalam program.

- g. *Testability*, usaha yang diperlukan dalam pengujian program untuk memastikan bahwa program melaksanakan fungsi yang ditetapkan.
- h. *Flexibility*, usaha yang diperlukan untuk memodifikasi program operasional.
- i. *Portability*, usaha yang diperlukan untuk memindahkan program dari perangkat keras / lingkungan sistem perangkat lunak tertentu ke yang lainnya.
- j. *Reusability*, tingkat kemampuan program / bagian dari program yang dapat dipakai ulang dalam aplikasi lainnya, berkaitan dengan paket dan lingkup dari fungsi yang dilakukan oleh program.
- k. *Interoperability*, usaha yang diperlukan untuk menggabungkan satu sistem dengan yang lainnya.

14. Pengujian *Software*

Pengujian merupakan metode yang dilakukan untuk menjelaskan tentang pengoperasian perangkat lunak yang terdiri dari perangkat pengujian, metode pengujian dan pelaksanaan pengujian. Pengujian terdiri atas 2 jenis, yaitu:

a. Pengujian *Alpha*

Pengujian *Alpha* dilakukan pada sisi pengembang. *Software* digunakan pada pengaturan yang natural dengan pengembang yg memandang sisi pemakai dan merekam semua kesalahan dan masalah pemakaian. Pengujian *Alpha* bertujuan untuk mengidentifikasi dan menghilangkan sebanyak mungkin masalah sebelum akhirnya sampai ke pengguna, dilakukan setelah *software* selesai oleh orang-orang

yang tidak terlibat dalam pengembangan dan memang ahli dibidangnya menggunakan formulir evaluasi resmi.

b. Pengujian *Betha*

Pengujian *Betha* dilakukan pada satu atau lebih pengguna *software* dalam lingkungan yg sebenarnya, pengembang tidak terlibat pengujian ini. Pengguna merekam semua masalah (*real* atau *imajiner*) yg ditemui selama pengujian dan melaporkan pada pengembang pada interval waktu tertentu. Berdasarkan hasil pengujian *betha*, dicari persentase masing-masing jawaban dengan menggunakan rumus:

$$Y = (P/Q) \times 100\%$$

Keterangan :

Y = Nilai persentase

P = Banyaknya jawaban responden tiap soal

Q = Jumlah responden

B. Kerangka Berpikir

Dalam proses bimbingan tugas akhir skripsi Jurdikmat UNY, terdapat beberapa kendala yang menyebabkan proses pengerjaan tugas akhir mahasiswa terhambat. Oleh karena itu, perlu dibuat sebuah sistem yang mampu mendukung proses bimbingan skripsi mahasiswa agar proses penyelesaian skripsi dapat lebih baik.

Di era teknologi informasi saat ini, semua kegiatan dilakukan dengan cepat, mudah, efektif dan efisien. Perguruan tinggi harus bisa memberikan kemudahan dalam berbagai macam informasi yang berkaitan dengan studi para mahasiswanya. Salah satu solusi untuk mendukung proses bimbingan skripsi mahasiswa adalah membuat Aplikasi Bimbingan Tugas Akhir Skripsi *Online* untuk mahasiswa, yaitu

sebuah sistem informasi berbasis *website* yang menyediakan informasi bagi mahasiswa yang ingin mengerjakan tugas akhir skripsi dan dilengkapi dengan fitur yang membantu proses bimbingan tugas akhir skripsi.

Aplikasi ini diharapkan akan memberikan kemudahan baik bagi para mahasiswa maupun dosen untuk menjalankan proses pembuatan tugas akhir skripsi. Dengan aplikasi ini, dokumentasi data skripsi akan lebih tertata dan mudah dilakukan dan memungkinkan mahasiswa untuk mencari informasi tentang judul skripsi, mengajukan proposal skripsi tanpa harus datang ke sekretariat jurusan, dan membantu mahasiswa dan dosen untuk berkomunikasi tanpa harus bertatap muka.

BAB III

PEMBAHASAN

A. Analisis Sistem

Sistem yang dianalisis adalah sistem pembuatan tugas akhir skripsi yang sedang berjalan di Jurdikmat UNY. Analisis ini dapat didefinisikan sebagai penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mempermudah mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi, dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya. Pada tahap analisis sistem harus dilakukan dengan teliti agar diketahui detail yang ada dalam sistem yang sedang berjalan saat ini. Analisis sistem dibagi menjadi 4 bagian, yaitu analisis masalah, analisis prosedur yang terlibat, analisis dokumen yang digunakan, dan analisis kebutuhan. Penjelasan lebih lanjut dari analisis-analisis tersebut adalah sebagai berikut:

1. Analisis Masalah

Sistem pembuatan tugas akhir skripsi yang sedang berjalan saat ini di Jurdikmat UNY memiliki beberapa kendala, sehingga perlu dibuat sebuah sistem yang mampu mengatasi kendala-kendala tersebut dan juga berguna untuk memudahkan dalam pengolahan data skripsi mahasiswa sehingga mendukung proses bimbingan tugas akhir skripsi mahasiswa agar proses penyelesaian tugas akhir skripsi bisa lebih baik.

2. Analisis Prosedur yang Terlibat

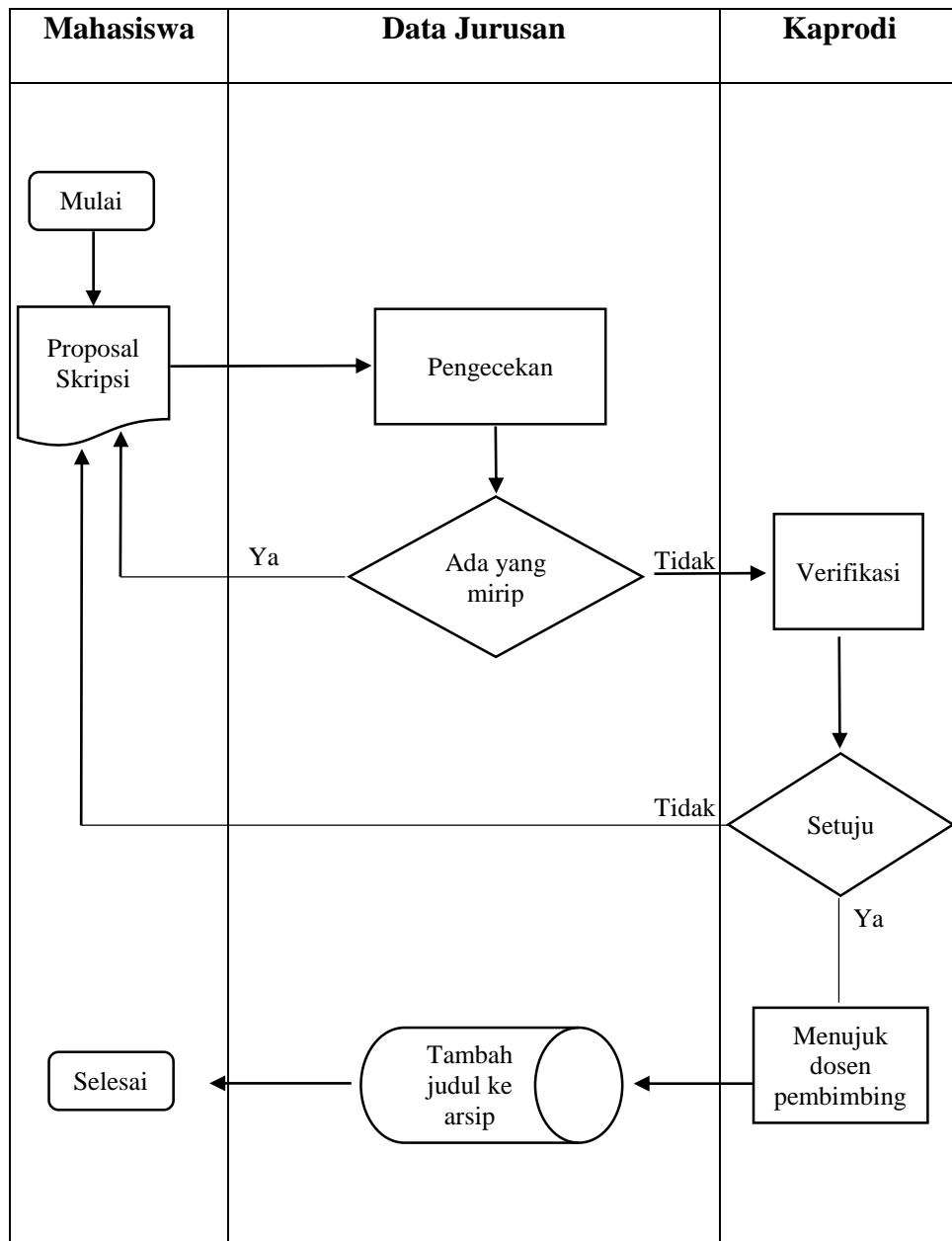
Prosedur merupakan urutan langkah-langkah yang terjadi atau yang dilakukan dalam suatu sistem. Prosedur yang terlibat dalam sistem pembuatan tugas akhir skripsi yang sedang berjalan saat ini di Jurdikmat UNY saat ini adalah sebagai berikut:

a. Prosedur Mengajukan Judul Skripsi

Prosedur mengajukan judul skripsi adalah prosedur dimana mahasiswa mengusulkan suatu masalah yang akan dibahas dalam skripsinya berupa proposal skripsi kepada Ketua Program Studi (Kaprodi) Jurdikmat UNY. Proses-proses dalam prosedur mengajukan judul skripsi adalah sebagai berikut:

- 1) Mahasiswa mengecek arsip judul skripsi yang sudah ada di sekretariat jurusan.
- 2) Jika tidak ditemui judul atau pokok bahasan yang mirip, proposal skripsi diajukan ke Kaprodi.
- 3) Kaprodi mengecek proposal skripsi. Jika diterima, judul ditambahkan ke arsip judul skripsi dan menunjuk dosen pembimbing bagi mahasiswa yang bersangkutan.

Flowmap dari prosedur ini disajikan pada Gambar 14.



Gambar 14. Flowmap Prosedur Mengajukan Judul Skripsi

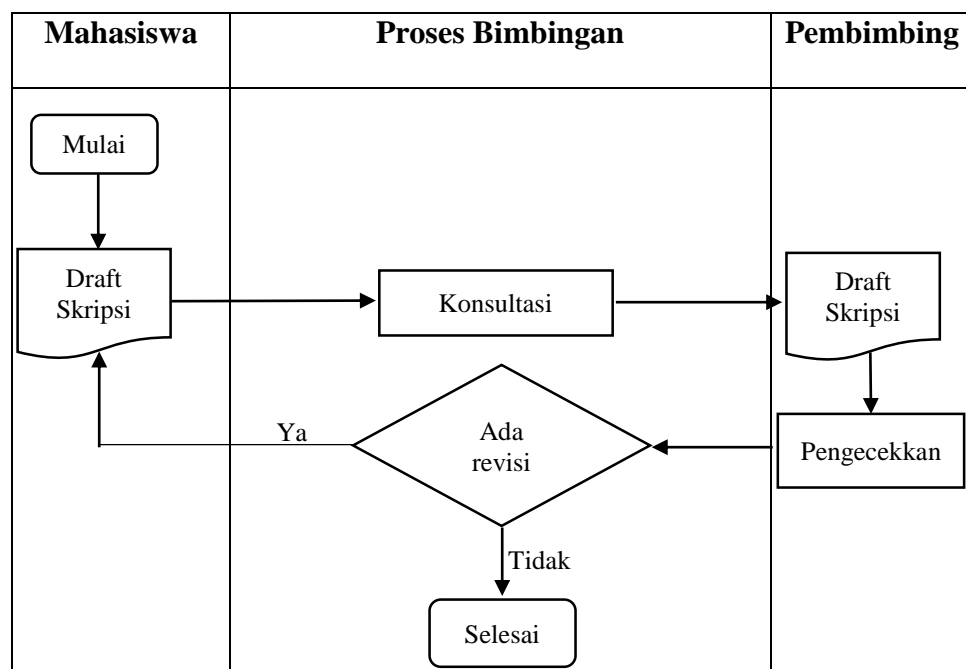
b. Prosedur Bimbingan Skripsi

Setelah kaprodi menentukan calon dosen pembimbing mahasiswa, kemudian mahasiswa yang bersangkutan meminta persetujuan dosen yang bersangkutan, ketua program studi, dan Ketua Jurdikmat (Kajurdik). Lalu mahasiswa mengajukan permohonan Surat Keputusan (SK) dosen pembimbing ke

sub bagian pendidikan FMIPA UNY. Setelah SK dosen pembimbing dibuat, mahasiswa dapat melakukan proses bimbingan skripsi. Prosedur bimbingan skripsi adalah kegiatan bimbingan skripsi yang dilakukan oleh mahasiswa dengan pembimbing skripsi. Proses-proses tersebut adalah sebagai berikut.

- 1) Mahasiswa mengajukan pertanyaan terkait hal-hal yang tidak/kurang dimengerti dalam pembuatan skripsinya termasuk isi skripsi yang telah dibuat (penulisan dan materi) sudah benar atau belum.
- 2) Dosen pembimbing menanggapi pertanyaan mahasiswa, mengevaluasi isi skripsi yang dibuat mahasiswa, dan memberikan bimbingan tentang isi skripsi yang benar.
- 3) Jika draft skripsi masih perlu revisi, kembali ke langkah 1.
- 4) Jika skripsi sudah disetujui dosen pembimbing, bimbingan skripsi selesai.

Flowmap dari prosedur ini disajikan pada Gambar 15.



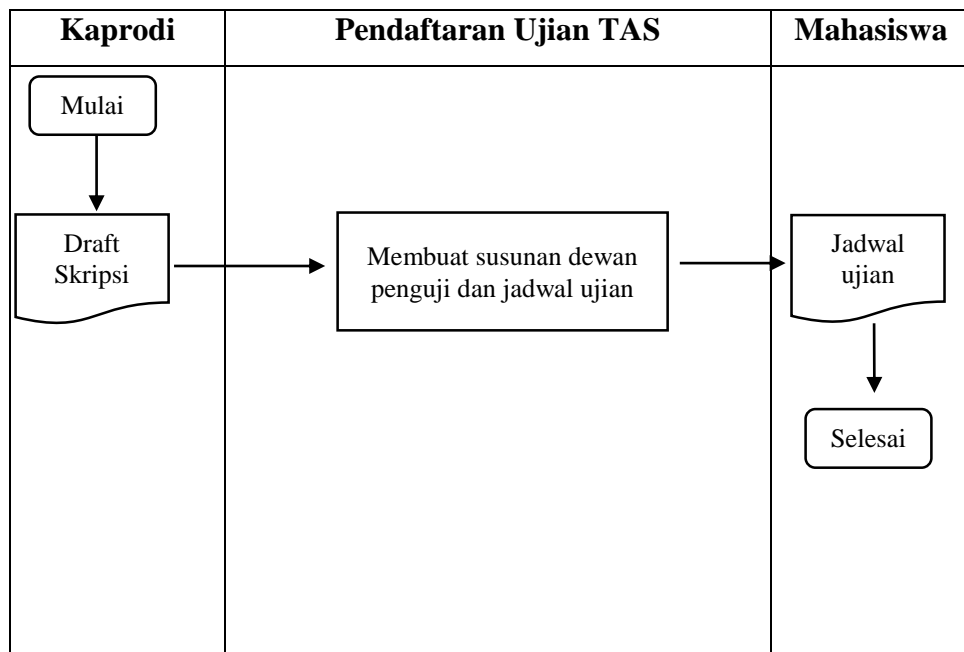
Gambar 15. Flowmap Prosedur Bimbingan Skripsi

c. Prosedur Pendaftaran Ujian Tugas Akhir Skripsi

Mahasiswa yang sudah selesai bimbingan skripsi dapat mendaftar ujian tugas akhir skripsi dengan datang ke jurusan/prodi membawa naskah yang sudah disetujui dosen pembimbing untuk diujikan dan telah memenuhi persyaratan ujian lainnya. Proses dalam prosedur pendaftaran ujian tugas akhir skripsi adalah sebagai berikut:

- 1) Kaprodi yang telah menerima naskah skripsi menentukan dewan penguji.
- 2) Kaprodi menentukan waktu dan tempat ujian, kemudian mengumumkannya.

Flowmap dari prosedur ini disajikan pada Gambar 16.



Gambar 16. Flowmap Prosedur Pendaftaran Ujian Tugas Akhir Skripsi

3. Analisis Dokumen yang Digunakan

Dokumen yang diolah dalam sistem pembuatan tugas akhir skripsi berasal dari prosedur yang dilakukan. Dokumen tersebut disajikan pada Tabel 1.

Tabel 1. Dokumen yang Digunakan Dalam Pembuatan Skripsi

No	Nama Dokumen	Deskripsi	Sumber
1	Draft skripsi	Berisi karya tulis mahasiswa, judul skripsi, dan data mahasiswa (NIM, nama)	Mahasiswa
2	Jadwal ujian	Berisi data dosen penguji, dan pernyataan penunjukkan penguji	Kaprodi
3	Arsip judul skripsi	Berisi daftar judul skripsi yang sudah digunakan/ dipakai	Sekretariat jurusan

Berdasarkan dokumen-dokumen di atas, maka informasi yang dihasilkan adalah sebagai berikut:

- a. Draft skripsi akan menghasilkan data NIM mahasiswa, nama mahasiswa, dan judul skripsi.
- b. Jadwal ujian akan menghasilkan data pada draft skripsi beserta data tanggal ujian, serta NIP dan nama dewan penguji.
- c. Arsip judul skripsi akan menghasilkan data pada jadwal ujian beserta data fokus studi, NIP dan nama pembimbing, dan tanggal pendaftaran.

4. Analisis Kebutuhan

Analisis kebutuhan merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi semua elemen dan komponen yang dibutuhkan untuk sistem yang akan dibangun, sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang dibutuhkan oleh sistem, keluaran yang akan dihasilkan oleh sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga

menghasilkan suatu keluaran yang diinginkan. Kebutuhan sistem terbagi menjadi beberapa analisis, yaitu sebagai berikut:

a. Analisis Perangkat Keras (*Hardware*)

Spesifikasi perangkat keras yang digunakan untuk membangun Aplikasi Bimbingan Tugas Akhir Skripsi *Online* tersebut adalah seperangkat komputer dengan spesifikasi sebagai berikut:

- 1) *Processor* Pentium IV 2,10 GHZ
- 2) *Hardisk* berkapasitas 249 GB
- 3) *Memory* berkapasitas 2,00 GB

b. Analisis Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan untuk membangun Aplikasi Bimbingan Tugas Akhir Skripsi *Online* tersebut adalah sebagai berikut:

- 1) Sistem operasi Windows 7.
- 2) *software Sublime 2.0* sebagai *text editor*.
- 3) *Database Management System* (DBMS) MySQL Xampp 7.
- 4) *Web browser Mozilla Firefox*.
- 5) *Framework Yii*.

c. Analisis Pengguna

Aplikasi yang dibuat dengan membatasi layanan–layanan yang disediakan berdasarkan hak akses yang diberikan kepada pengguna dalam lingkungan kampus sesuai keperluannya. Secara umum, pengguna memiliki hak untuk, *login*, melihat dan mengubah profil pribadinya, mengubah *password*, dan *logout*. Berdasarkan hak akses, pengguna yang menggunakan sistem Aplikasi Bimbingan Tugas Akhir

Skripsi *Online* yang akan dibuat dibedakan menjadi lima tipe pengguna, yaitu sebagai berikut:

1) Administrator

Administrator adalah staf Jurdikmat UNY yang diberikan kewenangan untuk melakukan pengelolaan (*maintenance*) aplikasi. Administrator dapat melihat dan mengubah semua data sistem, dan dapat membenahi kesalahan yang terjadi dalam sistem. Semua layanan yang ada pada sistem dapat digunakan oleh administrator dengan tujuan agar sistem bisa dimonitor secara penuh oleh administrator untuk perbaikan-perbaikan jika terjadi kesalahan yang dilakukan oleh pengguna baik secara disengaja maupun tidak disengaja.

2) Dosen

Dosen yang dimaksud adalah semua dosen aktif Jurdikmat UNY. Adapun hal-hal yang dapat dilakukan seorang dosen dalam sistem adalah sebagai berikut:

- a) Melihat daftar judul skripsi yang sudah digunakan mahasiswa Jurdikmat UNY.
- b) Melihat data dosen yang bersangkutan.
- c) Melihat daftar bimbingan.
- d) Mengirim pesan kepada mahasiswa bimbingannya.

3) Kaprodi

Kaprodi yang dimaksud adalah ketua program studi matematika dan ketua program studi pendidikan matematika Jurdikmat UNY yang sedang

bertugas. Adapun hal-hal yang dapat dilakukan seorang Kaprodi dalam sistem adalah sebagai berikut:

- a) Melihat daftar judul skripsi yang sudah digunakan mahasiswa Jurdikmat UNY.
- b) Melihat dan menyimpan proposal skripsi mahasiswa.
- c) Melakukan verifikasi kelayakan proposal skripsi yang diajukan, dan memberitahukan hasilnya kepada mahasiswa.
- d) Melihat data dosen.
- e) Menunjuk dosen pembimbing mahasiswa.
- f) Menentukan dewan penguji tugas akhir skripsi.
- g) Menentukan jadwal ujian tugas akhir skripsi dan memberitahukannya kepada dewan penguji dan mahasiswa yang bersangkutan.

4) Mahasiswa

Mahasiswa yang dimaksud adalah mahasiswa aktif Jurdikmat UNY yang telah menyelesaikan seluruh syarat pengambilan TA. Adapun hal-hal yang dapat dilakukan seorang mahasiswa dalam sistem adalah sebagai berikut:

- a) Mendaftarkan diri ke dalam sistem.
- b) Melihat daftar judul skripsi yang sudah digunakan mahasiswa Jurdikmat UNY lain.
- c) Mengajukan proposal skripsi kepada Kaprodi.
- d) Melihat daftar dosen.
- e) Konsultasi dengan dosen pembimbing.

5) Staf

Staf yang dimaksud adalah staf aktif Jurdikmat UNY yang mengelola kebutuhan administrasi tugas akhir skripsi mahasiswa Jurdikmat UNY. Adapun hal-hal yang dapat dilakukan seorang staf dalam sistem adalah sebagai berikut:

- a) Mendaftarkan pengguna.
- b) Melakukan pengumuman kepada seluruh pengguna.
- c) Menambah, atau mengubah data dosen.
- d) Menambah, atau mengubah mahasiswa terkait skripsi.

B. Perancangan Sistem

Perancangan sistem yang akan dibangun dimulai setelah tahap analisis terhadap sistem yang sedang berjalan selesai dilakukan. Perancangan dapat didefinisikan sebagai penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi. Alat bantu yang digunakan untuk menggambarkan perancangan sistem yang akan dibangun adalah sebagai berikut:

1. Perancangan Basis Data

Perancangan basis data adalah perancangan data yang tersimpan dalam media penyimpanan dan perangkat lunak yang akan memanipulasinya. Perancangan basis data terdiri dari struktur tabel dan skema relasi.

a. Struktur Tabel

Tabel-tabel yang digunakan dalam sistem bimbingan skripsi *online* terdiri dari 9 tabel. Berikut adalah struktur tabel-tabel tersebut:

1) Tabel *authassignment*

Tabel *authassignment* digunakan untuk menyimpan data hak akses pengguna yang telah ditentukan. Struktur tabel *authassignment* dapat dilihat pada Tabel 2.

Tabel 2. Struktur Tabel *authassignment*

No	Nama Field	Tipe data	Ukuran	Keterangan
1	<i>itemname</i>	<i>varchar</i>	64	Nama item (<i>Primary key</i>)
2	<i>userid</i>	<i>varchar</i>	64	Id pengguna (<i>Foreign key</i>)
3	<i>bizrule</i>	<i>text</i>	50	Skrip aturan
4	<i>data</i>	<i>Text</i>	50	Data item

2) Tabel *authitem*

Tabel *authitem* digunakan untuk menyimpan data tipe hak akses yang ada. Struktur tabel *authitem* dapat dilihat pada Tabel 3.

Tabel 3. Struktur Tabel *authitem*

No	Nama Field	Tipe data	Ukuran	Keterangan
1	<i>name</i>	<i>varchar</i>	64	Nama item (<i>Primary key</i>)
2	<i>type</i>	<i>int</i>	11	Tipe item
3	<i>description</i>	<i>text</i>	50	Keterangan tipe pengguna
4	<i>bizrule</i>	<i>text</i>	50	Skrip aturan
5	<i>data</i>	<i>text</i>	50	Data item

3) Tabel *authitemchild*

Tabel *authitemchild* digunakan untuk menyimpan data hubungan antar hak akses. Struktur tabel *authitemchild* dapat dilihat pada Tabel 4.

Tabel 4. Struktur Tabel authitemchild

No	Nama Field	Tipe data	Ukuran	Keterangan
1	<i>child</i>	<i>varchar</i>	64	Tipe pengguna (<i>Primary key</i>)
2	<i>parent</i>	<i>varchar</i>	64	Tipe pengguna (<i>Foreign key</i>)

4) Tabel dosen

Tabel dosen digunakan untuk menyimpan data dosen aktif yang ada di Jurdikmat UNY. Struktur tabel dosen dapat dilihat pada Tabel 5.

Tabel 5. Struktur Tabel dosen

No	Nama Field	Tipe data	Ukuran	Keterangan
1	id	<i>int</i>	10	Id data (<i>Primary key</i>)
2	nama	<i>varchar</i>	50	Nama dosen
3	nip	<i>int</i>	18	NIP dosen
4	tgl_lahir	<i>date</i>	8	Tanggal lahir dosen
5	jabatan	<i>varchar</i>	20	Jabatan dosen
6	pendidikan	<i>varchar</i>	100	Jenjang pendidikan dosen
7	bidang_keahlian	<i>varchar</i>	100	Keahlian dosen
8	pangkat	<i>varchar</i>	30	Pangkat dosen

5) Tabel mahasiswa

Tabel mahasiswa digunakan untuk menyimpan data mahasiswa yang telah memulai bimbingan. Struktur tabel mahasiswa dapat dilihat pada Tabel 6.

Tabel 6. Struktur Tabel mahasiswa

No	Nama <i>Field</i>	Tipe data	Ukuran	Keterangan
1	id	<i>int</i>	10	<i>Id data (Primary key)</i>
2	nim_mhs	<i>varchar</i>	11	NIM mahasiswa (<i>Unique</i>)
3	nama	<i>varchar</i>	50	Nama mahasiswa
4	fokus_studi	<i>varchar</i>	10	Fokus studi mahasiswa
5	judul_skripsi	<i>varchar</i>	250	Judul skripsi mahasiswa
6	pembimbing	<i>int</i>	18	NIP dosen (<i>Foreign key</i>)
7	tgl_masuk	<i>date</i>	8	Tanggal skripsi disetujui
8	tgl_ujian	<i>date</i>	8	Tanggal ujian skripsi
9	penguji1	<i>int</i>	18	NIP dosen (<i>Foreign key</i>)
10	penguji2	<i>int</i>	18	NIP dosen (<i>Foreign key</i>)
11	penguji3	<i>int</i>	18	NIP dosen (<i>Foreign key</i>)
12	penguji4	<i>int</i>	18	NIP dosen (<i>Foreign key</i>)

6) Tabel proposal

Tabel proposal digunakan untuk menyimpan keterangan proposal skripsi yang diajukan oleh mahasiswa. Struktur tabel proposal dapat dilihat pada Tabel 7.

Tabel 7. Struktur Tabel proposal

No	Nama <i>Field</i>	Tipe data	Ukuran	Keterangan
1	id_proposal	<i>int</i>	10	Id data (<i>Primary key</i>)
2	nim_mhs	<i>int</i>	11	NIM mahasiswa
3	judul_proposal	<i>varchar</i>	250	Judul skripsi yang diajukan (<i>Unique</i>)
4	file_proposal	<i>varchar</i>	30	Nama file yang di-upload

7) Tabel *mailbox conversation*

Tabel *mailbox conversation* digunakan untuk menyimpan data percakapan yang ditampilkan dalam menu pesan. Struktur tabel *mailbox conversation* dapat dilihat pada Tabel 8.

Tabel 8. Struktur Tabel *mailbox conversation*

No	Nama <i>Field</i>	Tipe data	Ukuran	Keterangan
1	<i>conversation_id</i>	<i>int</i>	10	Id <i>conversation</i> (<i>Primary key</i>)
2	<i>initiator_id</i>	<i>int</i>	10	Id <i>initiator</i>
3	<i>interlocutor_id</i>	<i>int</i>	10	Id <i>interlocutor</i>
4	<i>subject</i>	<i>varchar</i>	100	Subjek pesan
5	<i>bm_read</i>	<i>tinyint</i>	1	Status pesan terbaca
6	<i>bm_deleted</i>	<i>tinyint</i>	1	Status pesan terhapus
7	<i>modified</i>	<i>int</i>	10	Kode modifikasi
8	<i>is_system</i>	<i>enum</i>	1	Yes/No
9	<i>initiator_del</i>	<i>tinyint</i>	1	Status <i>initiator</i> terhapus
10	<i>interlocutor_del</i>	<i>tinyint</i>	1	Status <i>interlocutor</i> terhapus
11	<i>file</i>	<i>varchar</i>	30	Nama file yang di-upload

8) Tabel *mailbox message*

Tabel *mailbox message* digunakan untuk menyimpan data pesan. Struktur tabel *mailbox message* dapat dilihat pada Tabel 9.

Tabel 9. Struktur Tabel *mailbox message*

No	Nama <i>Field</i>	Tipe data	Ukuran	Keterangan
1	<i>message_id</i>	<i>int</i>	11	Id pesan (<i>Primary key</i>)
2	<i>conversation_id</i>	<i>int</i>	10	Id <i>conversation</i> (<i>Foreign key</i>)
3	<i>created</i>	<i>int</i>	10	Kode pembuatan
4	<i>sender_id</i>	<i>int</i>	10	<i>Username</i> pengirim
5	<i>recipient_id</i>	<i>int</i>	10	<i>Username</i> penerima
6	<i>text</i>	<i>text</i>	250	Isi pesan
7	<i>crc64</i>	<i>brigint</i>	20	Kode <i>crc64</i>
8	<i>file_foto</i>	<i>varchar</i>	30	Nama file yang di-upload

9) Tabel *users*

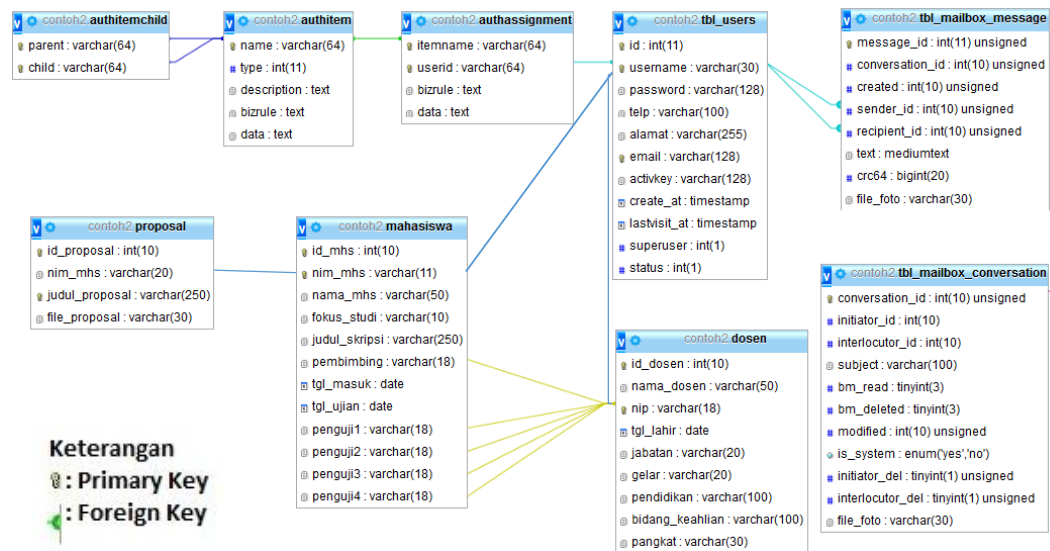
Tabel *users* digunakan untuk menyimpan data pengguna. Struktur tabel *users* dapat dilihat pada Tabel 10.

Tabel 10. Struktur Tabel *users*

No	Nama <i>Field</i>	Tipe data	Ukuran	Keterangan
1	<i>id</i>	<i>int</i>	10	Id data (<i>Primary key</i>)
2	<i>username</i>	<i>varchar</i>	30	NIM/NIP pengguna (<i>Unique</i>)
3	<i>password</i>	<i>varchar</i>	128	Password pengguna
4	<i>alamat</i>	<i>varchar</i>	100	Alamat pengguna
5	<i>telp</i>	<i>varchar</i>	255	Kontak pengguna
6	<i>nama</i>	<i>varchar</i>	128	Nama pengguna
7	<i>activekey</i>	<i>varchar</i>	128	Kunci aktivasi
8	<i>create_at</i>	<i>timestamp</i>	50	Tanggal pembuatan
9	<i>lastvisit_at</i>	<i>timestamp</i>	50	Tanggal terakhir login
10	<i>superuser</i>	<i>int</i>	1	Status superuser
11	<i>status</i>	<i>int</i>	1	Status akun

b. Skema Relasi

Skema relasi dari sistem yang akan dibangun disajikan pada Gambar 17. Tabel *users* berelasi dengan beberapa tabel yaitu tabel mahasiswa (untuk data NIM mahasiswa), tabel dosen (untuk data NIP dosen), *authassigment* (untuk data hak akses), dan *mailbox message* (untuk data *username* pengirim/penerima pesan). Tabel *authassigment* juga berelasi dengan tabel *authitem* (untuk data nama item). Tabel *authitem* juga berelasi dengan tabel *authitemchild* (untuk data nama item). Tabel mahasiswa juga berelasi dengan tabel dosen (untuk data NIP dosen pembimbing, penguji 1, penguji 2, penguji 3, dan penguji 4), dan tabel proposal (untuk data NIM mahasiswa). Tabel *mailbox message* juga berelasi dengan tabel *mailbox conversation* (untuk data *conservation*).



Gambar 17. Skema Relasi

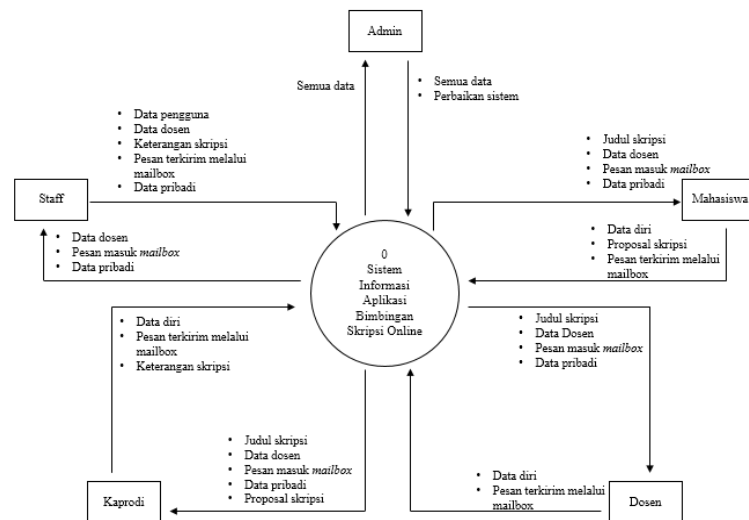
2. Perancangan Aliran Data

Perancangan aliran data adalah penggambaran dari aliran data yang masuk dan keluar dari sistem yang akan dibangun, aliran data dirancang menggunakan *Data Flow Diagram* (DFD).

Data Flow Diagram merupakan model sistem untuk menggambarkan pembagian sistem ke model yang lebih kecil secara paralel dan terstruktur dengan mengikutsertakan komponen-komponen dan entitas-entitas yang terkait, media penyimpanan (*Storage*), proses-proses sistem yang menunjukkan hubungan arus data dari proses ke entitas yang terkait. Salah satu keuntungan menggunakan DFD adalah memudahkan pemakai untuk memahami sistem yang sedang dibangun. DFD yang dibangun terdiri dari 3 jenis DFD, yaitu sebagai berikut:

a. Diagram Konteks

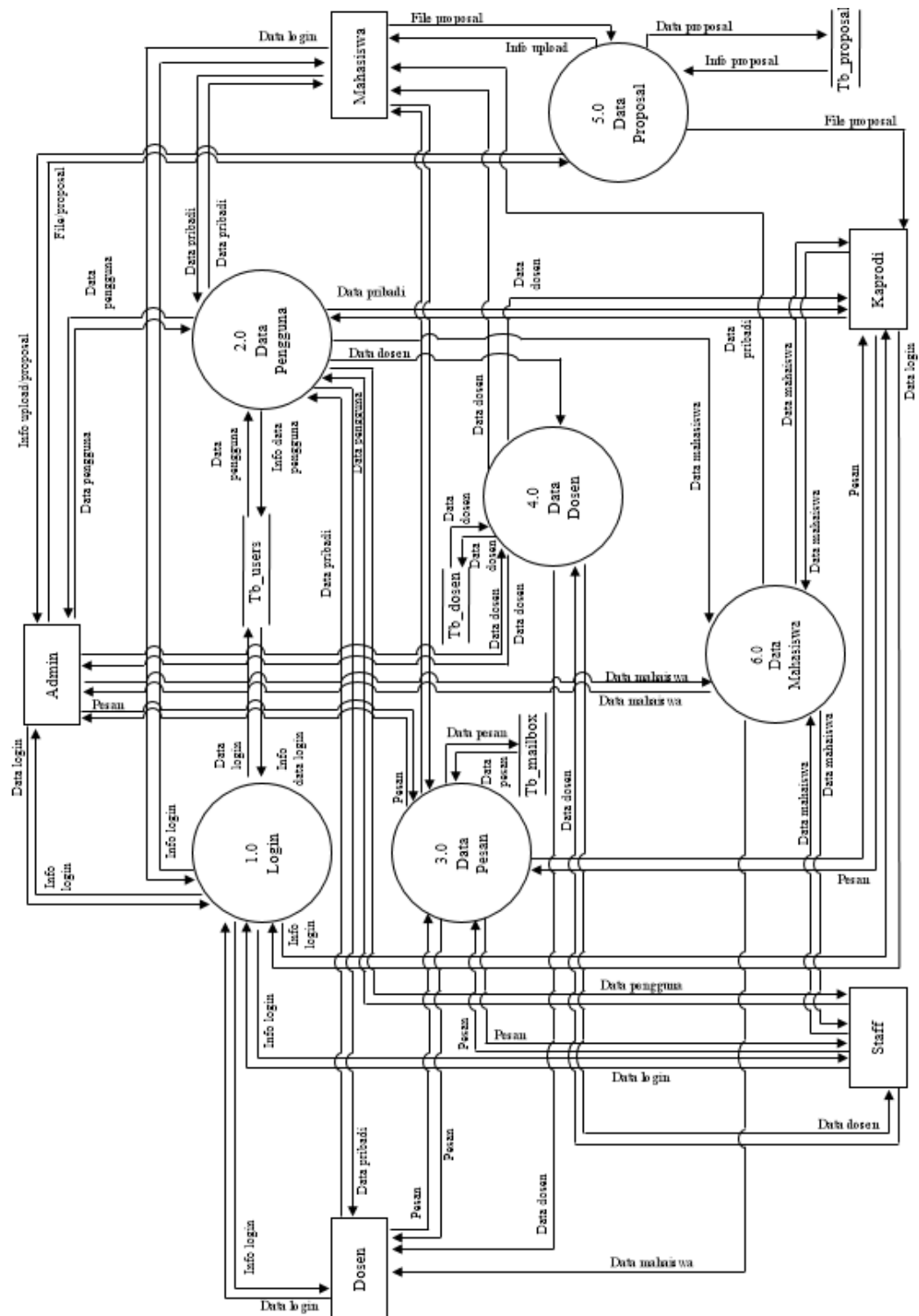
Diagram konteks merupakan alat untuk analisis struktur. Pendekatan struktur ini untuk menggambarkan sistem secara garis besar atau secara keseluruhan. Diagram konteks yang akan dibuat menghasilkan sumber informasi yang dibutuhkan dan tujuan yang akan dihasilkan. Diagram konteks dari sistem yang akan dibangun disajikan pada Gambar 18.



Gambar 18. Diagram Konteks Sistem Informasi Bimbingan Skripsi Online

b. DFD Level 1

DFD Level 1 menjelaskan gambaran program dan jalannya proses yang terjadi dalam sistem bimbingan skripsi *online* secara garis besar yang melibatkan entitas. Entitas yang terlibat dalam diagram konteks ada lima, yaitu administrator, dosen, kaprodi, mahasiswa, dan staff. Sedangkan proses yang terjadi dalam sistem terbagi menjadi 6 proses, yaitu proses *login*, pengolahan data pengguna, pengolahan data pesan, pengolahan data dosen, pengolahan data proposal, dan pengolahan data mahasiswa. Diagram konteks sistem bimbingan skripsi *online* disajikan pada Gambar 19.



Gambar 19. DFD Level 1 Sistem Informasi Bimbingan Skripsi Online

c. DFD Level 2 Proses 1.0 (Login)

Proses 1.0 merupakan pengolahan data *login* yang dilakukan oleh pengguna.

Proses login terdiri dari dua proses, yaitu sebagai berikut:

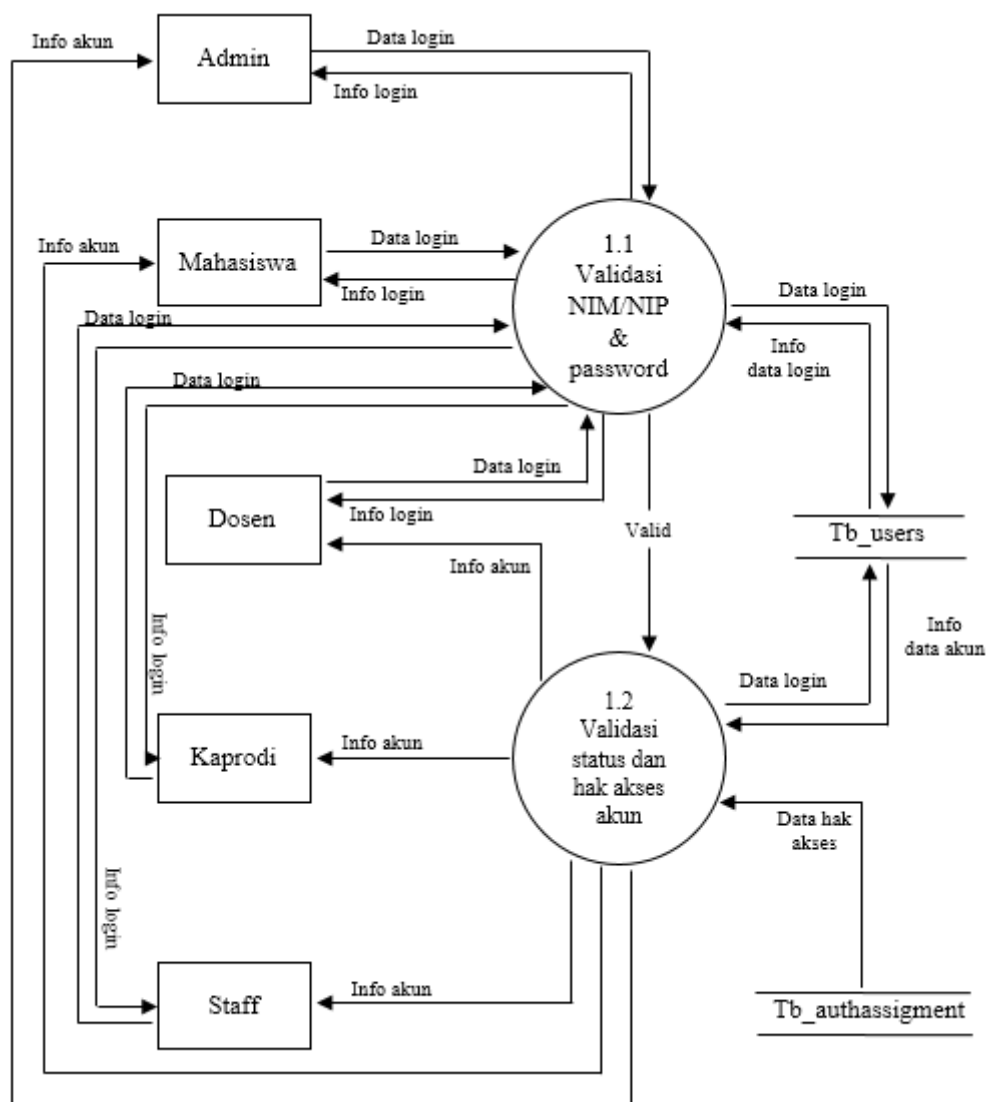
1) Proses validasi *username* dan *password*

Username dan *password* yang dimasukkan oleh pengguna saat *login* akan dicek pada tabel *users* keberadaan dan kecocokkannya. Jika *username* tidak ada atau tidak cocok dengan *password*, sistem akan menyampaikan pesan *error*. Jika *username* cocok dengan *password*, proses dilanjutkan ke proses validasi status pengguna.

2) Proses validasi status dan hak akses pengguna

Status pengguna yang melakukan proses *login* akan dicek pada tabel *users*, sedangkan hak akses pengguna akan dicek pada tabel *authassignment*. Status pengguna ada tiga, yaitu *active*, *not active*, dan *banned*. Jika status pengguna *active*, maka proses *login* berhasil. Untuk status pengguna selain *active*, proses *login* gagal dan pesan *error* ditampilkan.

DFD dari proses data *login* disajikan pada Gambar 20.



Gambar 20. DFD Level 2 Proses 1.0 (Login)

d. DFD Level 2 Proses 2.0 (Data Pengguna)

Proses 2.0 merupakan pengolahan data pengguna. Pengolahan data pengguna terdiri dari tiga proses, yaitu sebagai berikut:

1) Proses pendaftaran pengguna

Pengguna bisa mendaftarkan diri dengan memasukkan data diri ke dalam sistem. Data akan dimasukkan ke tabel *users* dengan status pengguna *not active*. Aktivasi pengguna bisa dilakukan oleh administrator atau staff.

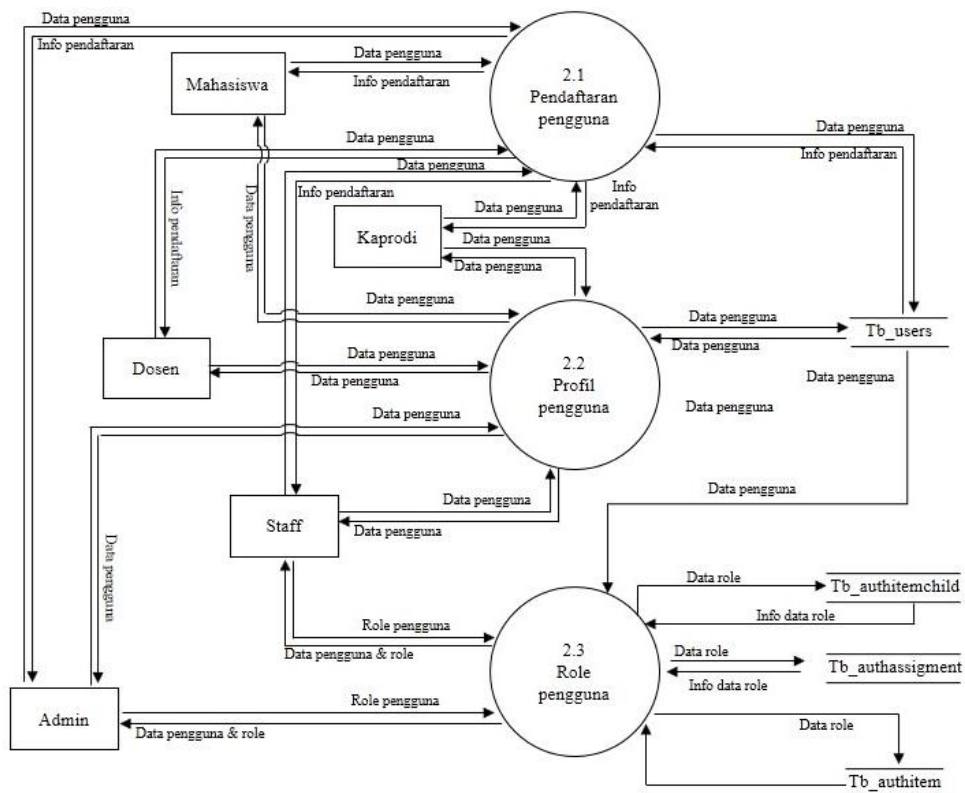
2) Proses pengolahan profil

Pengguna dapat mengubah beberapa data pribadinya. Data yang diubah akan disimpan di tabel *users* dan tabel *profiles* dengan mengganti data yang lama dengan yang data baru.

3) Proses pengolahan hak akses

Hak akses pengguna yang ada dalam sistem dibuat oleh administrator atau staff. Setiap pengguna memiliki satu hak akses yang ditentukan oleh administrator atau staff. Data hak akses disimpan di tabel *authassignment*, tabel *authitem*, dan tabel *authitemchild*.

DFD dari proses data pengguna disajikan pada Gambar 21.



Gambar 21. DFD Level 2 Proses 2.0 (Data Pengguna)

e. DFD Level 2 Proses 3.0 (Data Pesan)

Proses 3.0 merupakan proses pengolahan data pesan. Proses pengolahan data pesan terdiri dari empat proses, yaitu sebagai berikut:

1) Proses lihat pesan

Pengguna yang telah *login* bisa melihat pesan yang ada dengan mengakses menu pesan.

2) Proses buat pesan

Pengguna bisa membuat pesan kepada pengguna lain. Menu buat pesan terdiri dari 4 *field*, yaitu alamat tujuan yang diisi dengan *username* (NIM/NIP) pengguna lain, subjek pesan, *file attachment*, dan teks pesan.

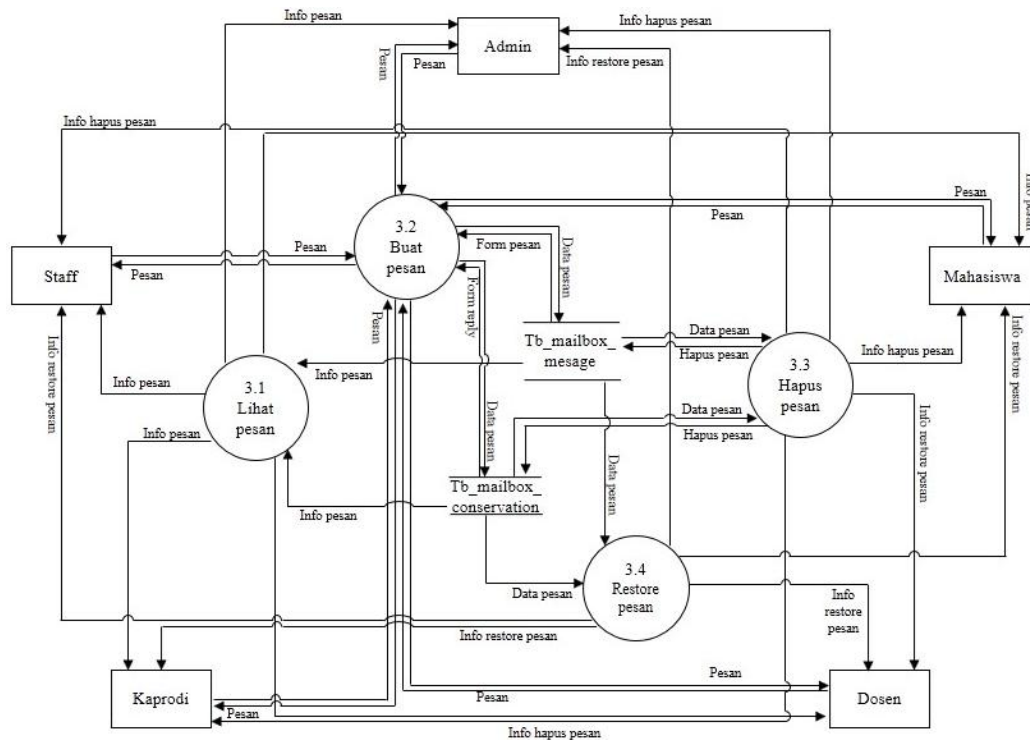
3) Proses hapus pesan

Pengguna bisa menghapus pesan dengan memilih menu hapus pada menu pesan. Pesan yang dihapus akan disimpan untuk selang waktu tertentu dan akan dihapus secara permanen setelah melewati selang waktu tersebut.

4) Proses *restore* pesan

Proses *restore* pesan adalah proses mengembalikan pesan yang telah dihapus tapi belum dihapus secara permanen. Pesan yang di-*restore* akan dapat dilihat kembali di menu pesan.

DFD dari proses data pesan disajikan pada Gambar 22.



f. DFD Level 2 Proses 4.0 (Data Dosen)

Proses 4.0 merupakan proses pengolahan data dosen. Data dosen dibuat oleh administrator atau staff dan disimpan pada tabel dosen. Proses pengolahan data dosen terdiri dari empat proses, yaitu sebagai berikut:

- 1) Proses lihat data

Pengguna yang telah *login* dapat melihat data dosen Jurdikmat UNY dengan mengakses menu dosen.

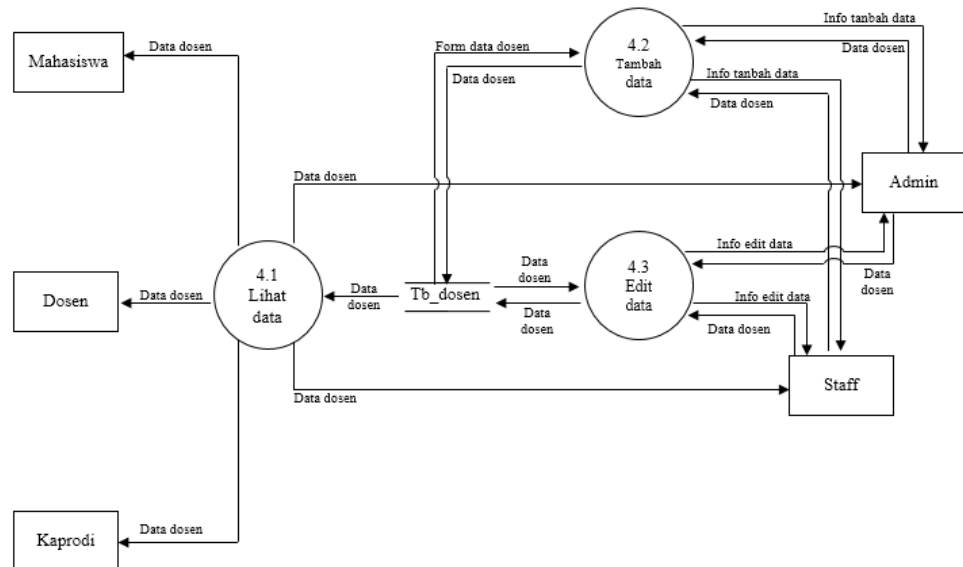
- ## 2) Proses tambah data

Data dosen hanya bisa dibuat atau ditambahkan oleh administrator atau staff dengan mengakses menu tambah data pada menu dosen.

3) Proses *edit* data

Data dosen hanya bisa diganti atau di-*edit* oleh administrator atau staff dengan mengakses menu *edit* data pada menu dosen.

DFD dari proses data dosen disajikan pada Gambar 23.



Gambar 23. DFD Level 2 Proses 4.0 (Data Dosen)

g. DFD Level 2 Proses 5.0 (Data Proposal)

Proses 5.0 merupakan proses pengolahan data proposal. Data proposal disimpan di tabel proposal. Proses data proposal terdiri dari tiga proses, yaitu sebagai berikut:

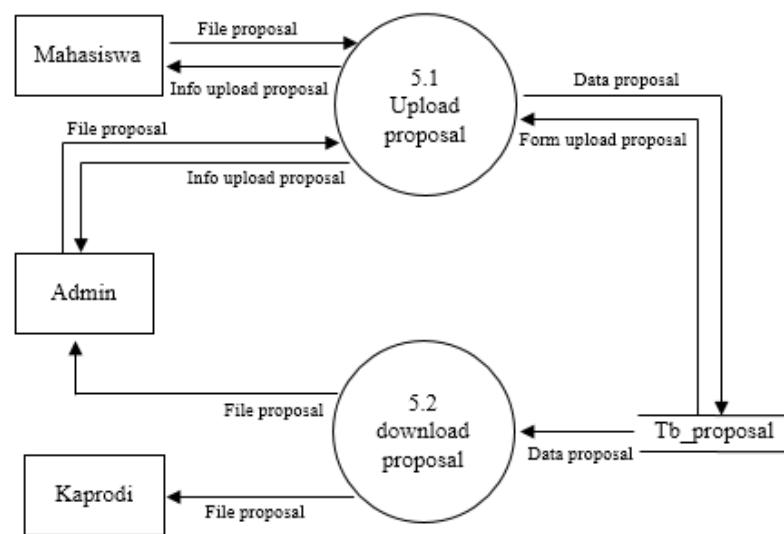
1) Proses *upload* proposal

Mahasiswa yang ingin mengajukan judul skripsi dapat meng-*upload file* proposal skripsinya di menu *upload* proposal.

2) Proses *download* proposal

File proposal dapat di-*download* oleh Kaprodi untuk dicek apakah layak atau tidak, kemudian hasilnya akan diberitahukan kepada Mahasiswa melalui pesan.

DFD dari proses data proposal disajikan pada Gambar 24.



Gambar 24. DFD Level 2 Proses 5.0 (Data Proposal)

h. DFD Level 2 Proses 6.0 (Data Mahasiswa)

Proses 6.0 adalah proses pengolahan data mahasiswa yang telah mengajukan judul skripsi dan telah memulai bimbingan skripsi. Data mahasiswa disimpan di tabel mahasiswa. Proses pengolahan data mahasiswa terdiri dari empat proses, yaitu sebagai berikut:

1) Proses lihat data

Pengguna yang telah *login* bisa melihat data mahasiswa yang telah mengajukan judul skripsi. Mahasiswa yang ingin mengajukan judul skripsi dapat melihat daftar judul skripsi yang telah digunakan di Jurdikmat UNY melalui menu ini.

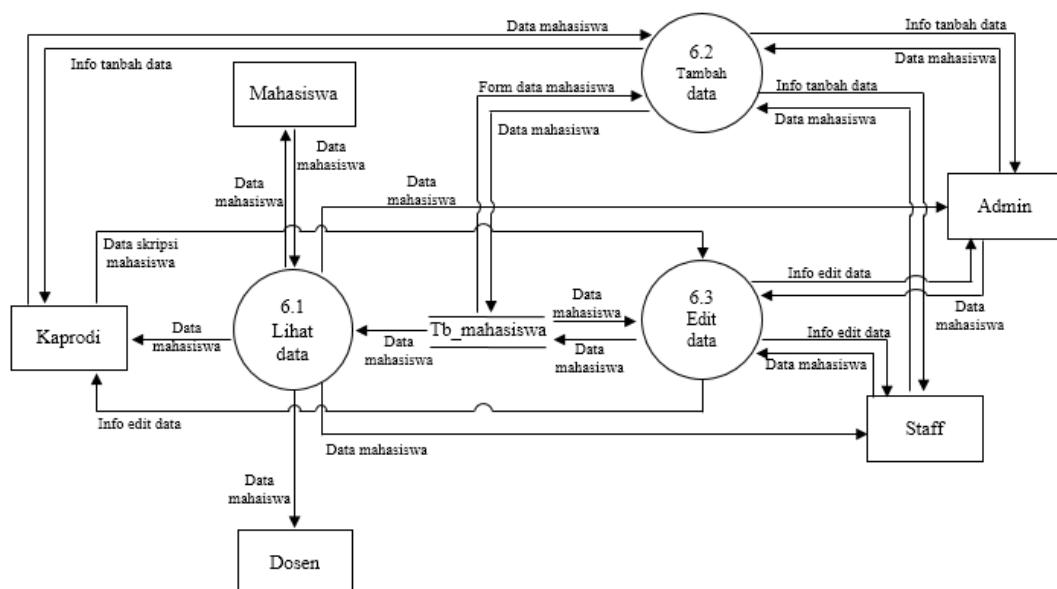
2) Proses tambah data

Data mahasiswa hanya bisa dibuat atau ditambahkan oleh administrator, staff, atau kaprodi dengan mengakses menu tambah data pada menu mahasiswa.

3) Proses *edit* data

Data mahasiswa hanya bisa diganti atau di-*edit* oleh administrator, staff, atau kaprodi atau kaprodi dengan mengakses menu *edit* data pada menu mahasiswa.

DFD dari proses data mahasiswa disajikan pada Gambar 25.



Gambar 25. DFD Level 2 Proses 6.0 (Data Mahasiswa)

i. DFD Level 3 Proses 2.2 (Profil Pengguna)

Proses 2.2 merupakan proses pengolahan profil pengguna. Data profil pengguna disimpan pada tabel *users* dan tabel *profiles*. Proses pengolahan data pengguna terdiri dari 3 proses, yaitu sebagai berikut:

1) Proses ubah data

Pengguna yang telah *login* dapat mengubah beberapa data pribadinya dengan mengakses menu ubah data pada menu profil.

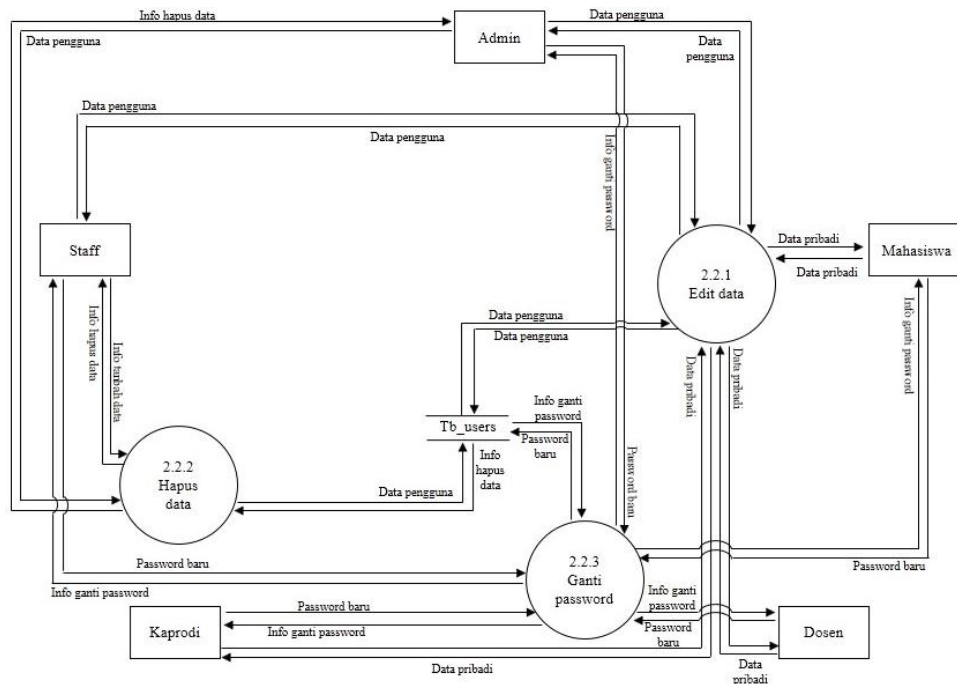
2) Proses hapus data

Proses ini adalah proses menghapus pengguna dari daftar pengguna dan hanya bisa dilakukan oleh administrator atau staff dengan mengakses menu hapus pengguna pada daftar pengguna.

3) Proses ganti password

Pengguna yang telah *login* dapat mengubah *password* dengan mengakses menu ubah *password* pada menu profil.

DFD dari proses data profil disajikan pada Gambar 26.



Gambar 26. DFD Level 3 Proses 2.2 (Profil Pengguna)

j. DFD Level 3 Proses 2.3 (Hak akses Pengguna)

Proses 2.3 merupakan proses pengolahan data hak akses pengguna. Data pengguna di akses melalui tabel *users* dan ditentukan hak aksesnya melalui menu hak akses. Data hak akses disimpan pada tabel *authassignment*, tabel *authitem*, dan tabel *authchild*. Proses pengolahan data hak akses terdiri dari empat proses, yaitu sebagai berikut:

1) Proses tambah hak akses

Proses ini hanya dapat dilakukan oleh administrator atau staff dengan mengakses menu tambah hak akses pada menu hak akses.

2) Proses *edit* hak akses

Proses ini hanya dapat dilakukan oleh administrator atau staff dengan mengakses menu *edit* hak akses pada menu hak akses.

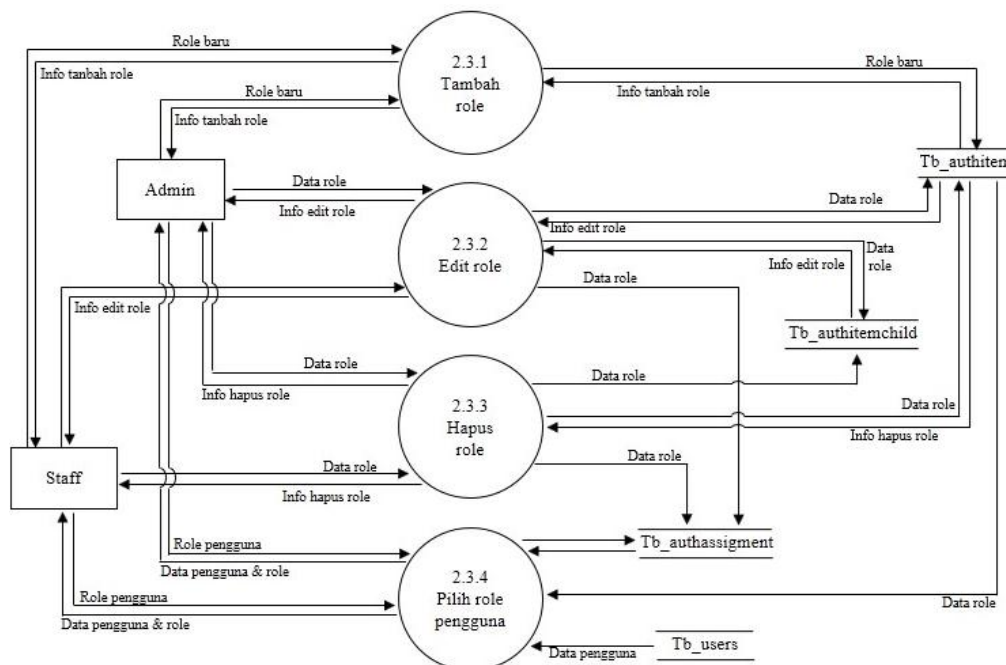
3) Proses hapus hak akses

Proses ini hanya dapat dilakukan oleh administrator atau staff dengan mengakses menu hapus hak akses pada menu hak akses.

4) Proses pilih hak akses pengguna

Proses ini hanya dapat dilakukan oleh administrator atau staff dengan mengakses menu hak akses.

DFD dari proses data hak akses disajikan pada Gambar 27.



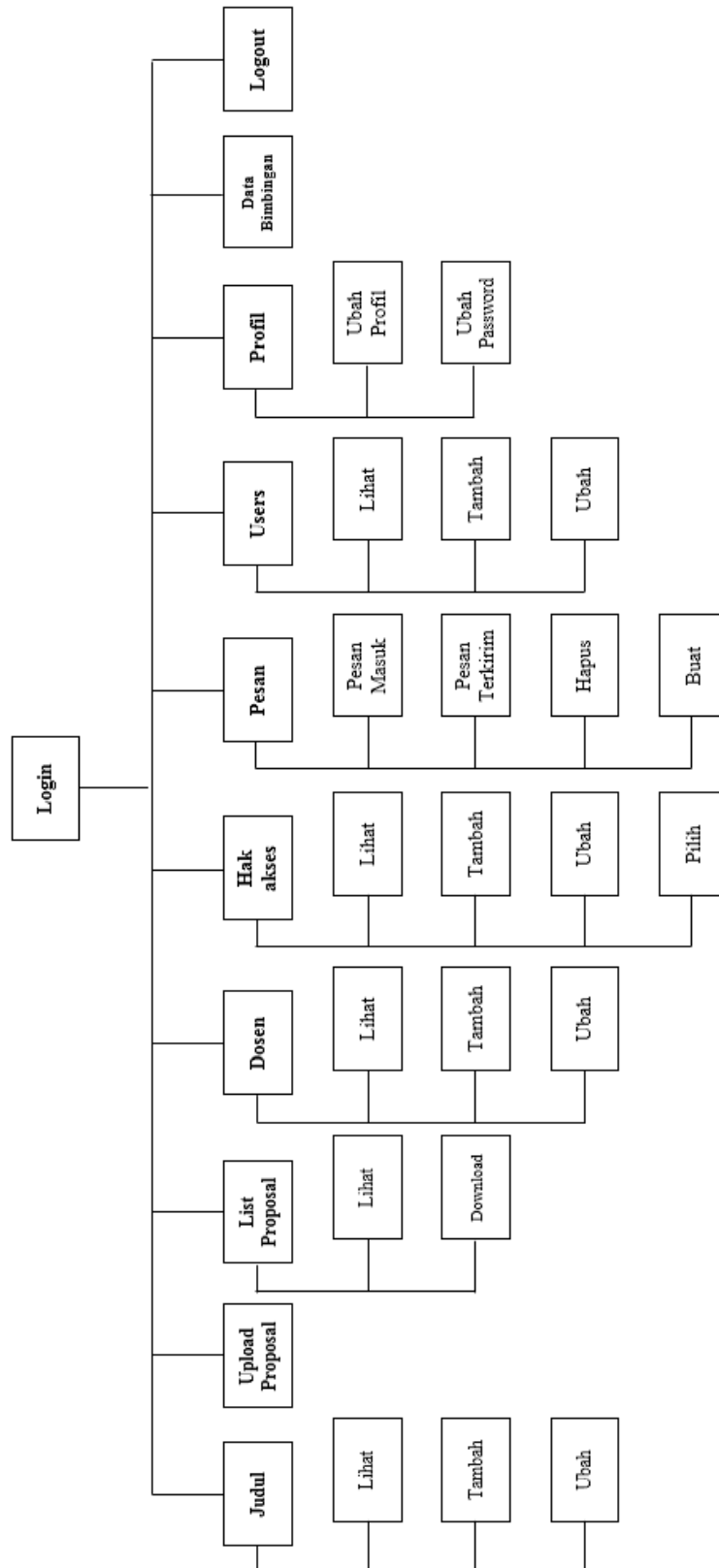
Gambar 27. DFD Level 3 Proses 2.3 (Hak akses Pengguna)

3. Perancangan Struktur Menu

Struktur menu dari sistem bimbingan skripsi *online* yang akan dibangun terdiri dari lima struktur menu berbeda berdasarkan hak akses pengguna.

a. Struktur Menu Administrator

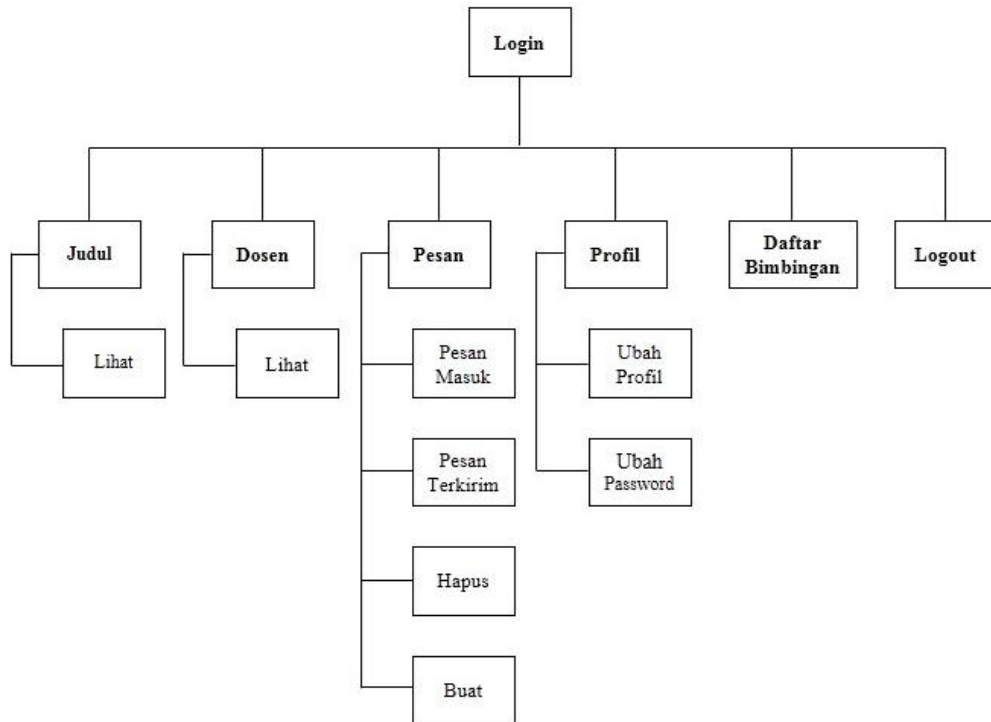
Administrator dapat mengakses semua menu yang terdiri dari tersebut adalah menu *login*, Judul, *Upload* Proposal, List Proposal, Dosen, Hak akses, Pesan, *Users*, profil, Daftar Bimbingan dan *logout*. Struktur menu Administrator disajikan pada Gambar 28.



Gambar 28. Struktur Menu Administrator

b. Struktur Menu Dosen

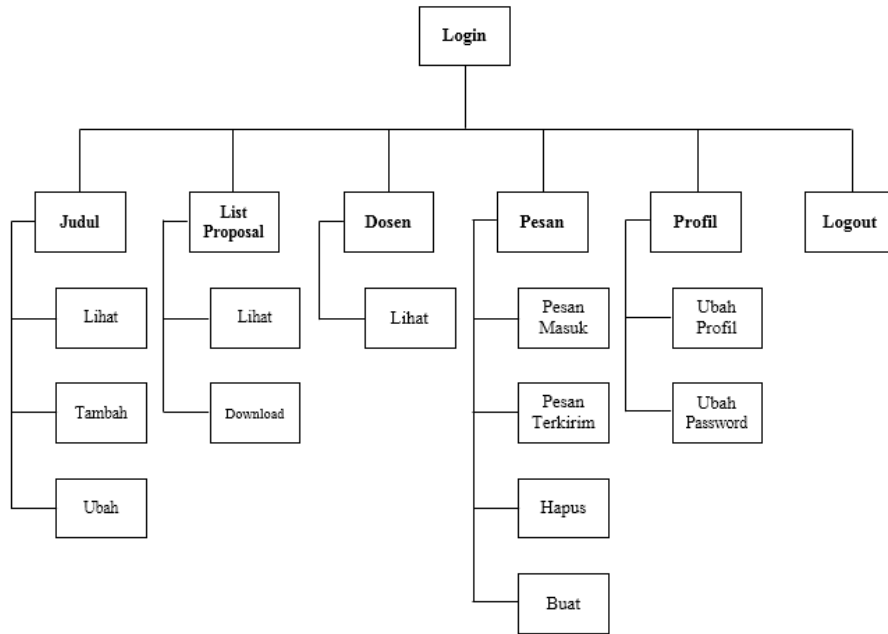
Dosen dapat mengakses tujuh menu yang ada pada sistem. Menu-menu tersebut adalah menu *login*, Judul, Dosen, Pesan, Profil, Daftar Bimbingan, dan *logout*. Struktur menu Dosen disajikan pada Gambar 29.



Gambar 29. Struktur Menu Dosen

c. Struktur Menu Kaprodi

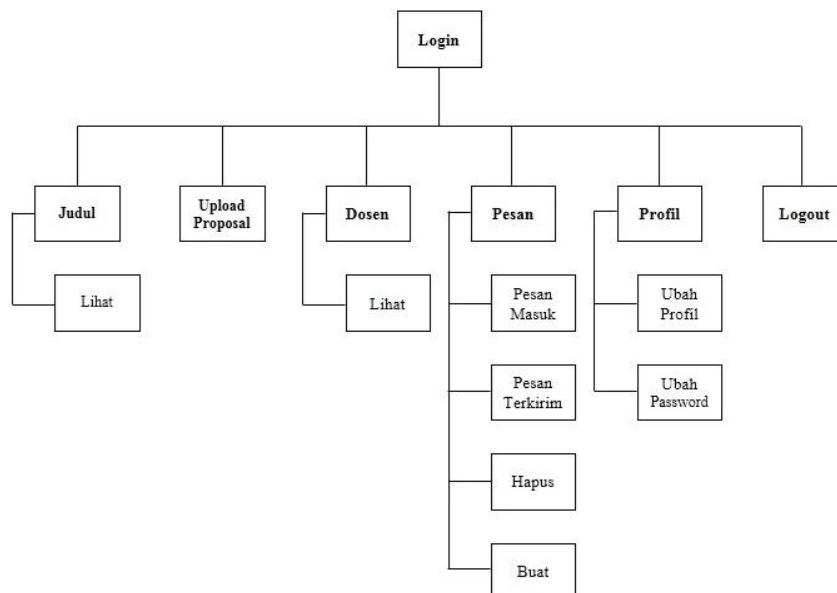
Kaprodi dapat mengakses tujuh menu yang ada pada sistem. Menu-menu tersebut adalah menu *login*, Judul (data mahasiswa), List Proposal, Dosen (data dosen), Pesan, profil, dan *logout*. Struktur menu Kaprodi disajikan pada Gambar 30.



Gambar 30. Struktur Menu Kaprodi

d. Struktur Menu Mahasiswa

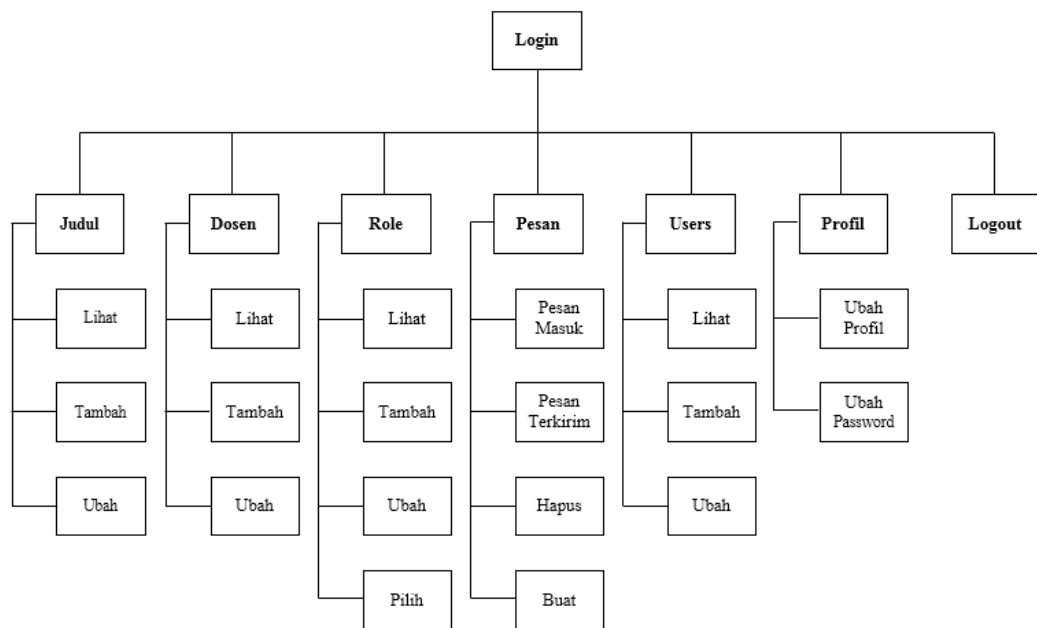
Mahasiswa dapat mengakses tujuh menu pada sistem. Menu-menu tersebut adalah menu *login*, *Judul* (data mahasiswa), *Upload Proposal*, *Dosen* (data dosen), *Pesan*, *profil*, dan *logout*. Struktur menu Mahasiswa disajikan pada Gambar 31.



Gambar 31. Struktur Menu Mahasiswa

e. Struktur Menu Staf

Staf dapat mengakses delapan menu yang ada pada sistem. Menu-menu tersebut adalah menu *login*, Judul (data mahasiswa), Dosen (data dosen), Hak akses, Pesan, *Users* (data *users*), profil, dan *logout*. Struktur menu staf disajikan pada Gambar 32.



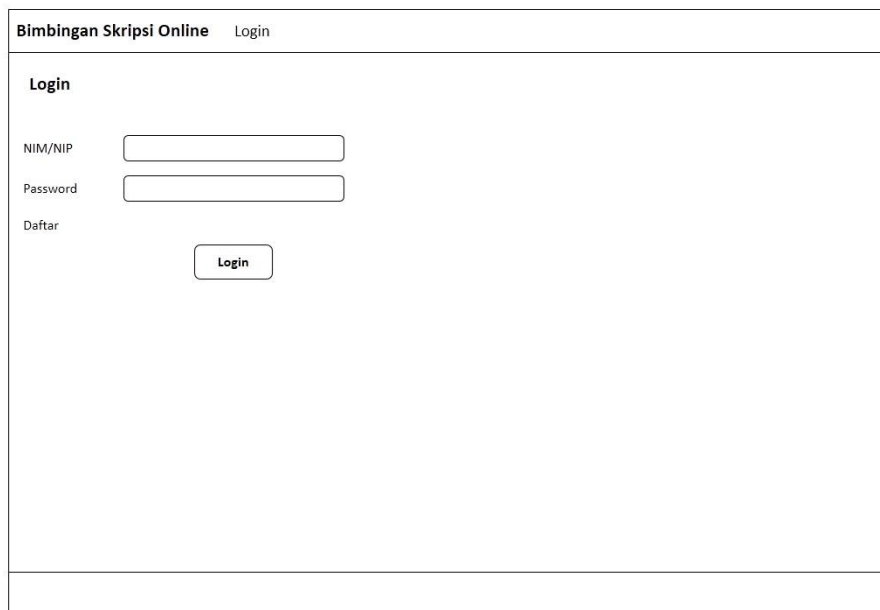
Gambar 32. Struktur Menu Staf

4. Perancangan Antar Muka

Rancangan antar muka aplikasi bimbingan skripsi *online* Jurdikmat UNY adalah sebagai berikut:

a. Rancangan Antar Muka Halaman *Login*

Halaman *login* adalah halaman pertama yang ditampilkan kepada pengguna yang belum *login* ketika mengakses aplikasi bimbingan skripsi *online*. Rancangan antar muka halaman *login* disajikan pada Gambar 33.

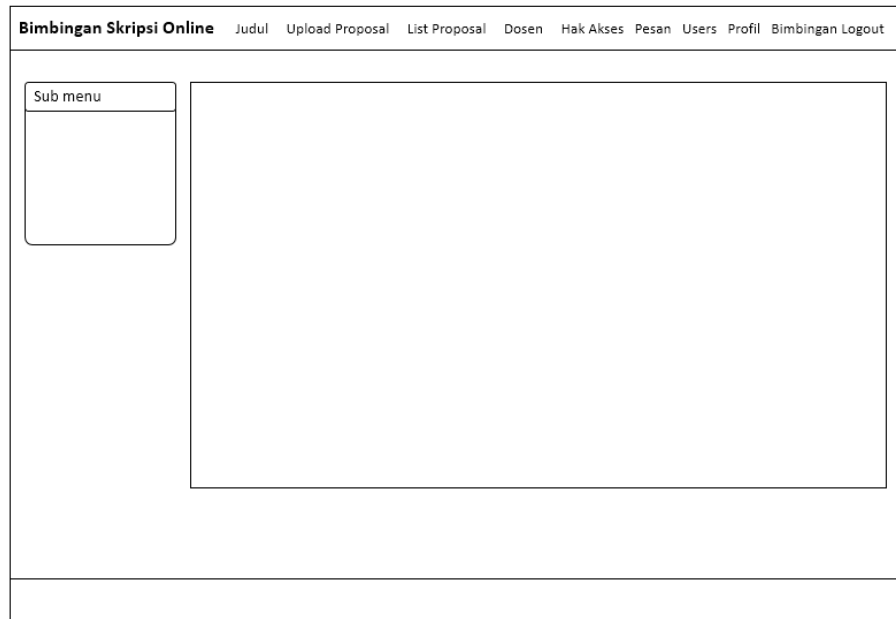


The image shows a login form for a system titled "Bimbingan Skripsi Online". At the top right of the header is a "Login" link. The main form area is titled "Login" and contains three input fields: "NIM/NIP", "Password", and "Daftar". Below the "Daftar" field is a "Login" button. The form is enclosed in a rectangular border.

Gambar 33. Rancangan Antar Muka Halaman Login

b. Rancangan Antar Muka Menu Pengguna (Administrator)

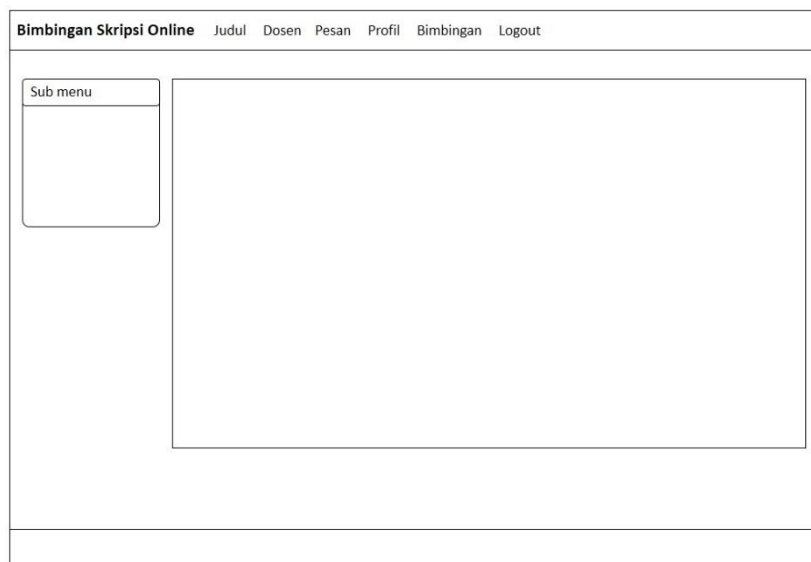
Pengguna dengan hak akses Administrator dapat mengakses sembilan menu setelah *login*, yaitu menu Judul, Upload Proposal, List Proposal, Dosen, Hak akses, Pesan, *Users*, Profil, dan *Logout*. Rancangan antar muka menu pengguna dengan hak akses Administrator disajikan pada Gambar 34.



Gambar 34. Rancangan Antar Muka Menu Pengguna (Administrator)

c. Rancangan Antar Muka Menu Pengguna (Dosen)

Pengguna dengan hak akses Dosen dapat mengakses lima menu setelah *login*, yaitu menu Judul, Dosen, Pesan, Profil, dan *Logout*. Rancangan antar muka menu pengguna dengan hak akses Dosen disajikan pada Gambar 35.

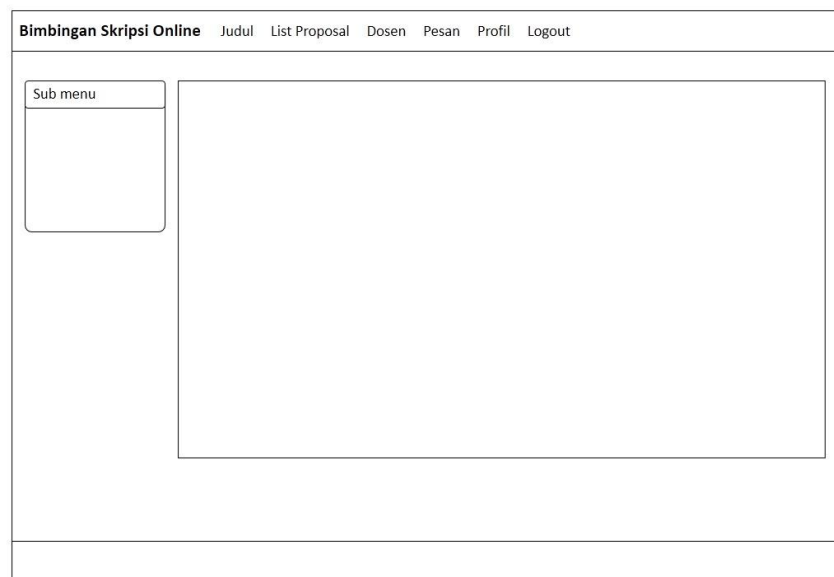


Gambar 35. Rancangan Antar Muka Menu Pengguna (Dosen)

d. Rancangan Antar Muka Menu Pengguna (Kaprodi)

Pengguna dengan hak akses Kaprodi dapat mengakses enam menu setelah *login*, yaitu menu Judul, List Proposal, Dosen, Pesan, Profil, dan *Logout*.

Rancangan antar muka menu pengguna dengan hak akses Kaprodi disajikan pada Gambar 36.

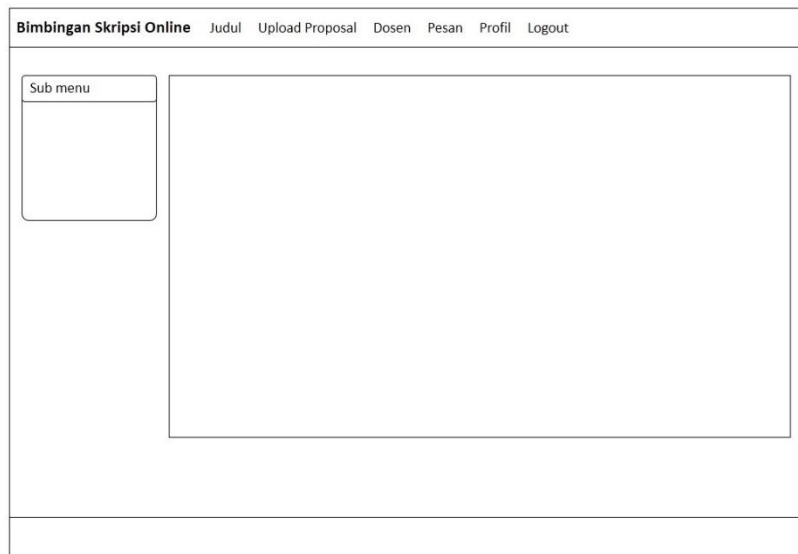


Gambar 36. Rancangan Antar Muka Menu Pengguna (Kaprodi)

e. Rancangan Antar Muka Menu Pengguna (Mahasiswa)

Pengguna dengan hak akses Mahasiswa dapat mengakses enam menu setelah *login*, yaitu menu Judul, Upload Proposal, Dosen, Pesan, Profil, dan *Logout*.

Rancangan antar muka menu pengguna dengan hak akses Mahasiswa disajikan pada Gambar 37.



Gambar 37. Rancangan Antar Muka Menu Pengguna (Mahasiswa)

f. Rancangan Antar Muka Menu Pengguna (Staf)

Pengguna dengan hak akses Staf dapat mengakses tujuh menu setelah *login*, yaitu menu Judul, Dosen, Hak akses, Pesan, *Users*, Profil, dan *Logout*. Rancangan antar muka menu pengguna dengan hak akses Mahasiswa disajikan pada Gambar 38.



Gambar 38. Rancangan Antar Muka Menu Pengguna (Staf)

g. Rancangan Antar Muka Menu Judul

Menu Judul digunakan untuk mengelola dan menampilkan data mahasiswa Jurdikmat UNY yang telah mengajukan judul skripsi. Rancangan antar muka menu Judul disajikan pada Gambar 39, Gambar 40, Gambar 41, Gambar 42, dan Gambar 43.

Bimbingan Skripsi Online																																																						
Judul Upload Proposal List Proposal Dosen Hak Akses Pesan Users Profil Bimbingan logout																																																						
Sub menu * Tambah data		Daftar Skripsi Mahasiswa <table border="1"> <thead> <tr> <th>NIM</th> <th>Nama</th> <th>Judul</th> <th>Tgl Masuk</th> <th>Tgl Ujian</th> <th>Opsi</th> </tr> </thead> <tbody> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>Detail/Ubah</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>Detail/Ubah</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>Detail/Ubah</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>Detail/Ubah</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>Detail/Ubah</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>Detail/Ubah</td></tr> </tbody> </table>					NIM	Nama	Judul	Tgl Masuk	Tgl Ujian	Opsi												Detail/Ubah						Detail/Ubah						Detail/Ubah						Detail/Ubah						Detail/Ubah						Detail/Ubah
NIM	Nama	Judul	Tgl Masuk	Tgl Ujian	Opsi																																																	
					Detail/Ubah																																																	
					Detail/Ubah																																																	
					Detail/Ubah																																																	
					Detail/Ubah																																																	
					Detail/Ubah																																																	
					Detail/Ubah																																																	

Gambar 39. Rancangan Antar Muka Menu Judul (Administrator)

Bimbingan Skripsi Online																																																						
Judul Dosen Pesan Profil Bimbingan logout																																																						
Daftar Skripsi Mahasiswa <table border="1"> <thead> <tr> <th>NIM</th> <th>Nama</th> <th>Judul</th> <th>Tgl Masuk</th> <th>Tgl Ujian</th> <th>Opsi</th> </tr> </thead> <tbody> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>Detail</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>Detail</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>Detail</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>Detail</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>Detail</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>Detail</td></tr> </tbody> </table>							NIM	Nama	Judul	Tgl Masuk	Tgl Ujian	Opsi												Detail						Detail						Detail						Detail						Detail						Detail
NIM	Nama	Judul	Tgl Masuk	Tgl Ujian	Opsi																																																	
					Detail																																																	
					Detail																																																	
					Detail																																																	
					Detail																																																	
					Detail																																																	
					Detail																																																	

Gambar 40. Rancangan Antar Muka Menu Judul (Dosen)

Bimbingan Skripsi Online Judul List Proposal Dosen Pesan Profil Logout

Sub menu
* Tambah data

Daftar Skripsi Mahasiswa

NIM	Nama	Judul	Tgl Masuk	Tgl Ujian	Opsi
					Detail/Ubah
					Detail/Ubah
					Detail/Ubah
					Detail/Ubah
					Detail/Ubah
					Detail/Ubah

Gambar 41. Rancangan Antar Muka Menu Judul (Kaprodi)

Bimbingan Skripsi Online Judul Upload Proposal Dosen Pesan Profil Logout

Daftar Skripsi Mahasiswa

NIM	Nama	Judul	Tgl Masuk	Tgl Ujian	Opsi
					Detail
					Detail
					Detail
					Detail
					Detail
					Detail

Gambar 42. Rancangan Antar Muka Menu Judul (Mahasiswa)

Bimbingan Skripsi Online
Judul
Dosen
Hak Akses
Pesan
Users
Profil
Logout

Sub menu
* Tambah data

Daftar Skripsi Mahasiswa

NIM	Nama	Judul	Tgl Masuk	Tgl Ujian	Opsi
					Detail/Ubah
					Detail/Ubah
					Detail/Ubah
					Detail/Ubah
					Detail/Ubah
					Detail/Ubah

Gambar 43. Rancangan Antar Muka Menu Judul (Staf)

Rancangan antar muka form tambah data Judul disajikan pada Gambar 44, dan Gambar 45.

Bimbingan Skripsi Online
Judul
Upload Proposal
List Proposal
Dosen
Hak Akses
Pesan
Users
Profil
Bimbingan
logout

Sub menu
* Lihat Daftar

Tambah Judul Skripsi

NIM
Nama
Fokus Studi
Judul Skripsi
Pembimbing
Tanggal Masuk
Tanggal Ujian
Ketua Penguji
Sekretaris Penguji
Penguji Utama
Penguji Pendamping

Tambah

Gambar 44. Rancangan Antar Muka Form Tambah Data Judul (Administrator)

Bimbingan Skripsi Online		Judul	Dosen	Hak Akses	Pesan	Users	Profil	Logout
<div>Sub menu</div> <div>* Lihat Daftar</div>		<div>Tambah Judul Skripsi</div> <div> <div>NIM</div> <div>Nama</div> <div>Fokus Studi</div> <div>Judul Skripsi</div> <div>Pembimbing</div> <div>Tanggal Masuk</div> <div>Tanggal Ujian</div> <div>Ketua Penguji</div> <div>Sekretaris Penguji</div> <div>Penguji Utama</div> <div>Penguji Pendamping</div> </div> <div>Tambah</div>						

Gambar 45. Rancangan Antar Muka Form Tambah Data Judul (Staf)

Rancangan antar muka detail data Judul disajikan pada Gambar 46, Gambar 47, Gambar 48, Gambar 49, dan Gambar 50.

Bimbingan Skripsi Online		Judul	Upload Proposal	List Proposal	Dosen	Hak Akses	Pesan	Users	Profil	Bimbingan	logout
<div>Sub menu</div> <div> * Tambah data * Ubah data * Lihat Daftar </div>		<div>Detail Judul</div> <div> <div>NIM</div> <div>Nama</div> <div>Fokus Studi</div> <div>Judul Skripsi</div> <div>Pembimbing</div> <div>Tanggal Masuk</div> <div>Tanggal Ujian</div> <div>Ketua Penguji</div> <div>Sekretaris Penguji</div> <div>Penguji Utama</div> <div>Penguji Pendamping</div> </div>									

Gambar 46. Rancangan Antar Muka Detail Data Judul (Administrator)

Bimbingan Skripsi Online		Judul	Dosen	Pesan	Profil	Bimbingan	Logout
<div>Sub menu</div> <div>* Lihat Daftar</div>		<div>Detail Judul</div> <div> NIM : Nama : Fokus Studi : Judul Skripsi : Pembimbing : Tanggal Masuk : Tanggal Ujian : Ketua Penguji : Sekretaris Penguji : Penguji Utama : Penguji Pendamping : </div>					

Gambar 47. Rancangan Antar Muka Detail Data Judul (Dosen)

Bimbingan Skripsi Online		Judul	List Proposal	Dosen	Pesan	Profil	Logout
<div>Sub menu</div> <div> * Tambah data * Ubah data * Lihat Daftar </div>		<div>Detail Judul</div> <div> NIM : Nama : Fokus Studi : Judul Skripsi : Pembimbing : Tanggal Masuk : Tanggal Ujian : Ketua Penguji : Sekretaris Penguji : Penguji Utama : Penguji Pendamping : </div>					

Gambar 48. Rancangan Antar Muka Detail Data Judul (Kaprod)

Bimbingan Skripsi Online		Judul	Upload Proposal	Dosen	Pesan	Profil	Logout
Sub menu							
* Lihat Daftar							
<div> <div>Detail Judul</div> <div> NIM : Nama : Fokus Studi : Judul Skripsi : Pembimbing : Tanggal Masuk : Tanggal Ujian : Ketua Penguji : Sekretaris Penguji : Penguji Utama : Penguji Pendamping : </div> </div>							

Gambar 49. Rancangan Antar Muka Detail Data Judul (Mahasiswa)

Bimbingan Skripsi Online		Judul	Dosen	Hak Akses	Pesan	Users	Profil	Logout
Sub menu								
* Tambah data * Ubah data * Lihat Daftar								
<div> <div>Detail Judul</div> <div> NIM : Nama : Fokus Studi : Judul Skripsi : Pembimbing : Tanggal Masuk : Tanggal Ujian : Ketua Penguji : Sekretaris Penguji : Penguji Utama : Penguji Pendamping : </div> </div>								

Gambar 50. Rancangan Antar Muka Detail Data Judul (Staf)

h. Rancangan Antar Muka Menu Upload Proposal

Menu *Upload* Proposal digunakan oleh pengguna dengan hak akses Mahasiswa untuk mengajukan proposal skripsi kepada Kaprodi dengan meng-*upload* file proposal skripsinya. Rancangan antar muka menu *Upload* Proposal disajikan pada Gambar 51 dan Gambar 52.

Gambar 51. Rancangan Antar Muka Menu Upload Proposal (Administrator)

Gambar 52. Rancangan Antar Muka Menu Upload Proposal (Mahasiswa)

i. Rancangan Antar Muka Menu List Proposal

Menu List Proposal digunakan oleh pengguna dengan hak akses Kaprodi untuk men-*download file* proposal skripsi yang diajukan (di-*upload*) mahasiswa. Rancangan menu *List Proposal* disajikan pada Gambar 53 dan Gambar 54.

Bimbingan Skripsi Online	Judul	Upload Proposal	List Proposal	Dosen	Hak Akses	Pesan	Users	Profil	Bimbingan	logout
---------------------------------	-------	-----------------	---------------	-------	-----------	-------	-------	--------	-----------	--------

Daftar Proposal			
NIM Mahasiswa	Judul Proposal	File Proposal	Opsi
			Detail
			Detail
			Detail
			Detail
			Detail
			Detail
			Detail

Gambar 53. Rancangan Antar Muka Menu List Proposal (Administrator)

Bimbingan Skripsi Online	Judul	List Proposal	Dosen	Pesan	Profil	Logout
---------------------------------	-------	---------------	-------	-------	--------	--------

Daftar Proposal			
NIM Mahasiswa	Judul Proposal	File Proposal	Opsi
			Detail
			Detail
			Detail
			Detail
			Detail
			Detail
			Detail

Gambar 54. Rancangan Antar Muka Menu List Proposal (Kaprodi)

j. Rancangan Antar Muka Daftar Dosen

Daftar Dosen digunakan untuk melihat dan mengelola data dosen Jurdikmat UNY. Rancangan antar muka menu dosen disajikan pada Gambar 55, Gambar 56, Gambar 57, Gambar 58, dan Gambar 59.

Bimbingan Skripsi Online Judul Upload Proposal List Proposal Dosen Hak Akses Pesan Users Profil Bimbingan logout

Sub menu
* Tambah data

Daftar Dosen

Nama	NIP	Tgl Lahir	Bidang Keahlian	Jabatan	Opsi
					Detail/Ubah
					Detail/Ubah
					Detail/Ubah
					Detail/Ubah
					Detail/Ubah
					Detail/Ubah

Gambar 55. Rancangan Antar Muka Daftar Dosen (Administrator)

Bimbingan Skripsi Online Judul Dosen Pesan Profil Bimbingan Logout

Daftar Dosen

Nama	NIP	Tgl Lahir	Bidang Keahlian	Jabatan	Opsi
					Detail
					Detail
					Detail
					Detail
					Detail
					Detail

Gambar 56. Rancangan Antar Muka Daftar Dosen (Dosen)

Bimbingan Skripsi Online Judul List Proposal Dosen Pesan Profil Logout

Daftar Dosen

Nama	NIP	Tgl Lahir	Bidang Keahlian	Jabatan	Opsi
					Detail
					Detail
					Detail
					Detail
					Detail
					Detail

Gambar 57. Rancangan Antar Muka Daftar Dosen (Kaprodi)

Bimbingan Skripsi Online Judul Upload Proposal Dosen Pesan Profil Logout

Daftar Dosen

Nama	NIP	Tgl Lahir	Bidang Keahlian	Jabatan	Opsi
					Detail
					Detail
					Detail
					Detail
					Detail
					Detail

Gambar 58. Rancangan Antar Muka Daftar Dosen (Mahasiswa)

Bimbingan Skripsi Online																																																												
Judul	Dosen	Hak Akses	Pesan	Users	Profil	Logout																																																						
<div style="border: 1px solid black; padding: 5px;">Sub menu</div> <div>* Tambah data</div>																																																												
<div style="border: 1px solid black; padding: 10px;"> <p>Daftar Dosen</p> <table border="1"> <thead> <tr> <th>Nama</th> <th>NIP</th> <th>Tgl Lahir</th> <th>Bidang Keahlian</th> <th>Jabatan</th> <th>Opsi</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td>Detail/Ubah</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td>Detail/Ubah</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td>Detail/Ubah</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td>Detail/Ubah</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td>Detail/Ubah</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td>Detail/Ubah</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td>Detail/Ubah</td></tr> </tbody> </table> </div>							Nama	NIP	Tgl Lahir	Bidang Keahlian	Jabatan	Opsi												Detail/Ubah						Detail/Ubah						Detail/Ubah						Detail/Ubah						Detail/Ubah						Detail/Ubah						Detail/Ubah
Nama	NIP	Tgl Lahir	Bidang Keahlian	Jabatan	Opsi																																																							
					Detail/Ubah																																																							
					Detail/Ubah																																																							
					Detail/Ubah																																																							
					Detail/Ubah																																																							
					Detail/Ubah																																																							
					Detail/Ubah																																																							
					Detail/Ubah																																																							

Gambar 59. Rancangan Antar Muka Daftar Dosen (Staf)

Rancangan antar muka detail data dosen disajikan pada Gambar 60, Gambar 61, Gambar 62, Gambar 63, dan Gambar 64.

Bimbingan Skripsi Online	
Judul	Upload Proposal
List Proposal	Dosen
Hak Akses	Pesan
Users	Profil
Bimbingan	Logout

Sub menu

- * Tambah data
- * Ubah data
- * Lihat Daftar

Detail Dosen

Nama :
NIP :
Tanggal Lahir :
Jabatan :
Pendidikan :
Bidang Keahlian :
Pangkat :

Gambar 60. Rancangan Antar Muka Detail Data Dosen (Administrator)

Bimbingan Skripsi Online		Judul	Dosen	Pesan	Profil	Bimbingan	Logout
<div>Sub menu</div> <div>* Lihat Daftar</div>	<div>Detail Dosen</div> <div> Nama : NIP : Tanggal Lahir : Jabatan : Pendidikan : Bidang Keahlian : Pangkat : </div>						

Gambar 61. Rancangan Antar Muka Detail Data Dosen (Dosen)

Bimbingan Skripsi Online		Judul	List Proposal	Dosen	Pesan	Profil	Logout
<div>Sub menu</div> <div>* Lihat Daftar</div>	<div>Detail Dosen</div> <div> Nama : NIP : Tanggal Lahir : Jabatan : Pendidikan : Bidang Keahlian : Pangkat : </div>						

Gambar 62. Rancangan Antar Muka Detail Data Dosen (Kaprodi)

Bimbingan Skripsi Online		Judul	Upload Proposal	Dosen	Pesan	Profil	Logout
<div>Sub menu</div> <div>* Lihat Daftar</div>	<div>Detail Dosen</div> <div> Nama : NIP : Tanggal Lahir : Jabatan : Pendidikan : Bidang Keahlian : Pangkat : </div>						

Gambar 63. Rancangan Antar Muka Detail Data Dosen (Mahasiswa)

Bimbingan Skripsi Online		Judul	Dosen	Hak Akses	Pesan	Users	Profil	Logout
<div>Sub menu</div> <div> * Tambah data * Ubah data * Lihat Daftar </div>	<div>Detail Dosen</div> <div> Nama : NIP : Tanggal Lahir : Jabatan : Pendidikan : Bidang Keahlian : Pangkat : </div>							

Gambar 64. Rancangan Antar Muka Detail Data Dosen (Staf)

k. Rancangan Antar Muka Menu Hak akses

Menu Hak Akses digunakan oleh Administrator atau Staf untuk mengelola hak akses pengguna. Pengguna yang baru membuat atau dibuatkan akun, hak akses pengguna tersebut akan ditentukan melalui menu Hak Akses sebelum akun

pengguna tersebut diaktivasi. Rancangan antar muka menu Hak Akses disajikan pada Gambar 65 dan Gambar 66.

Bimbingan Skripsi Online																			
Judul Upload Proposal List Proposal Dosen Hak Akses Pesan Users Profil Bimbingan logout																			
Sub menu * Daftar Role	Daftar Hak Akses Pengguna <table border="1"> <thead> <tr> <th>NIM/NIP</th> <th>Pilih Role</th> </tr> </thead> <tbody> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> </tbody> </table>	NIM/NIP	Pilih Role																
NIM/NIP	Pilih Role																		

Gambar 65. Rancangan Antar Muka Menu Hak Akses (Administrator)

Bimbingan Skripsi Online																			
Judul Dosen Hak Akses Pesan Users Profil Logout																			
Sub menu * Daftar Role	Daftar Hak Akses Pengguna <table border="1"> <thead> <tr> <th>NIM/NIP</th> <th>Pilih Role</th> </tr> </thead> <tbody> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> </tbody> </table>	NIM/NIP	Pilih Role																
NIM/NIP	Pilih Role																		

Gambar 66. Rancangan Antar Muka Menu Hak Akses (Staf)

Rancangan antar muka daftar Hak Akses disajikan pada Gambar 67 dan Gambar 68.

Bimbingan Skripsi Online Judul Upload Proposal List Proposal Dosen Hak Akses Pesan Users Profil Bimbingan logout

Sub menu
* Tambah Role

Daftar Hak Akses

Hak Akses	Opsi
	Detail/Ubah
	Detail/Ubah
	Detail/Ubah
	Detail/Ubah
	Detail/Ubah
	Detail/Ubah

Gambar 67. Rancangan Antar Muka Daftar Hak Akses (Administrator)

Bimbingan Skripsi Online Judul Dosen Hak Akses Pesan Users Profil Logout

Sub menu
* Tambah Role

Daftar Hak Akses

Hak Akses	Opsi
	Detail/Ubah
	Detail/Ubah
	Detail/Ubah
	Detail/Ubah
	Detail/Ubah
	Detail/Ubah

Gambar 68. Rancangan Antar Muka Daftar Hak Akses (Staf)

1. Rancangan Antar Muka Menu Pesan

Menu Pesan digunakan oleh pengguna untuk mengirim pesan ke pengguna lain. Menu ini digunakan untuk melakukan kegiatan bimbingan skripsi. Rancangan

antar muka menu Pesan disajikan Gambar 69, Gambar 70, Gambar 71, Gambar 72, dan Gambar 73.

Bimbingan Skripsi Online Judul Upload Proposal List Proposal Dosen Hak Akses Pesan Users Profil Bimbingan logout	
<div>Pesan Masuk</div> <div>Pesan Terkirim</div> <div>Pesan Terhapus</div> <div>Buat Pesan</div>	<div>Pesan</div> <div>Pesan</div> <div>Pesan</div>

Gambar 69. Rancangan Antar Muka Menu Pesan (Administrator)

Bimbingan Skripsi Online Judul Dosen Pesan Profil Bimbingan Logout	
<div>Pesan Masuk</div> <div>Pesan Terkirim</div> <div>Pesan Terhapus</div> <div>Buat Pesan</div>	<div>Pesan</div> <div>Pesan</div> <div>Pesan</div>

Gambar 70. Rancangan Antar Muka Menu Pesan (Dosen)

Bimbingan Skripsi Online		Judul	List Proposal	Dosen	Pesan	Profil	Logout
<div>Pesan Masuk</div> <div>Pesan Terkirim</div> <div>Pesan Terhapus</div> <div>Buat Pesan</div>	<div>Pesan</div> <div>Pesan</div> <div>Pesan</div>						

Gambar 71. Rancangan Antar Muka Menu Pesan (Kaprodi)

Bimbingan Skripsi Online		Judul	Upload Proposal	Dosen	Pesan	Profil	Logout
<div>Pesan Masuk</div> <div>Pesan Terkirim</div> <div>Pesan Terhapus</div> <div>Buat Pesan</div>	<div>Pesan</div> <div>Pesan</div> <div>Pesan</div>						

Gambar 72. Rancangan Antar Muka Menu Pesan (Mahasiswa)

Bimbingan Skripsi Online		Judul	Dosen	Hak Akses	Pesan	Users	Profil	Logout
<div>Pesan Masuk</div> <div>Pesan Terkirim</div> <div>Pesan Terhapus</div> <div>Buat Pesan</div>	<div>Pesan</div> <div>Pesan</div> <div>Pesan</div>							

Gambar 73. Rancangan Antar Muka Menu Pesan (Staf)

Rancangan antar muka menu Buat Pesan disajikan pada Gambar 74, Gambar 75, Gambar 76, Gambar 77, dan Gambar 78.

Bimbingan Skripsi Online		Judul	Upload Proposal	List Proposal	Dosen	Hak Akses	Pesan	Users	Profil	Bimbingan	logout
<div>Pesan Masuk</div> <div>Pesan Terkirim</div> <div>Pesan Terhapus</div> <div>Buat Pesan</div>	<div> <div>Kepada</div> <div>Subjek</div> <div>Attachment</div> <div>Isi Pesan</div> <div>Kirim Pesan</div> </div> <div> <div>Browse</div> </div>										

Gambar 74. Rancangan Antar Muka Buat Pesan (Administrator)

Bimbingan Skripsi Online
[Judul](#)
[Dosen](#)
[Pesan](#)
[Profil](#)
[Bimbingan](#)
[Logout](#)

Pesan Masuk

Pesan Terkirim

Pesan Terhapus

Buat Pesan

Kepada

Subjek

Attachment

Isi Pesan

Kirim Pesan

Browse

Gambar 75. Rancangan Antar Muka Buat Pesan (Dosen)

Bimbingan Skripsi Online
[Judul](#)
[List Proposal](#)
[Dosen](#)
[Pesan](#)
[Profil](#)
[Logout](#)

Pesan Masuk

Pesan Terkirim

Pesan Terhapus

Buat Pesan

Kepada

Subjek

Attachment

Isi Pesan

Kirim Pesan

Browse

Gambar 76. Rancangan Antar Muka Buat Pesan (Kaprod)

Bimbingan Skripsi Online

[Judul](#)
[Upload Proposal](#)
[Dosen](#)
[Pesan](#)
[Profil](#)
[Logout](#)

Pesan Masuk

Pesan Terkirim

Pesan Terhapus

Buat Pesan

Kepada

Subjek

Attachment

Isi Pesan

Kirim Pesan

Browse

Gambar 77. Rancangan Antar Muka Buat Pesan (Mahasiswa)

Bimbingan Skripsi Online

[Judul](#)
[Dosen](#)
[Hak Akses](#)
[Pesan](#)
[Users](#)
[Profil](#)
[Logout](#)

Pesan Masuk

Pesan Terkirim

Pesan Terhapus

Buat Pesan

Kepada

Subjek

Attachment

Isi Pesan

Kirim Pesan

Browse

Gambar 78. Rancangan Antar Muka Buat Pesan (Staf)

Rancangan antar muka menu Baca Pesan disajikan pada Gambar 79, Gambar 80, Gambar 81, Gambar 82, dan Gambar 83.

Bimbingan Skripsi Online		Judul	Upload Proposal	List Proposal	Dosen	Hak Akses	Pesan	Users	Profil	Bimbingan	logout
<div>Pesan Masuk</div> <div>Pesan Terkirim</div> <div>Pesan Terhapus</div> <div>Buat Pesan</div>	<div>Subjek</div> <div>Attachment</div> <div>Pengirim/ Anda</div> <div>Pengirim/ Anda</div> <div>Pengirim/ Anda</div> <div>Balas Pesan</div> <div>Kirim Pesan</div>										

Gambar 79. Rancangan Antar Muka Baca Pesan (Administrator)

Bimbingan Skripsi Online		Judul	Dosen	Pesan	Profil	Bimbingan	Logout
<div>Pesan Masuk</div> <div>Pesan Terkirim</div> <div>Pesan Terhapus</div> <div>Buat Pesan</div>	<div>Subjek</div> <div>Attachment</div> <div>Pengirim/ Anda</div> <div>Pengirim/ Anda</div> <div>Pengirim/ Anda</div> <div>Balas Pesan</div> <div>Kirim Pesan</div>						

Gambar 80. Rancangan Antar Muka Baca Pesan (Dosen)

Bimbingan Skripsi Online Judul List Proposal Dosen Pesan Profil Logout

Pesan Masuk

Pesan Terkirim

Pesan Terhapus

Buat Pesan

Subjek

Attachment

Pengirim/ Anda

Pengirim/ Anda

Pengirim/ Anda

Balas Pesan

Kirim Pesan

Gambar 81. Rancangan Antar Muka Baca Pesan (Kaprodi)

Bimbingan Skripsi Online Judul Upload Proposal Dosen Pesan Profil Logout

Pesan Masuk

Pesan Terkirim

Pesan Terhapus

Buat Pesan

Subjek

Attachment

Pengirim/ Anda

Pengirim/ Anda

Pengirim/ Anda

Balas Pesan

Kirim Pesan

Gambar 82. Rancangan Antar Muka Baca Pesan (Mahasiswa)

Bimbingan Skripsi Online Judul Dosen Hak Akses Pesan Users Profil Logout

Pesan Masuk
Pesan Terkirim
Pesan Terhapus
Buat Pesan

Subjek
Attachment
Pengirim/Anda
Pengirim/Anda
Pengirim/Anda
Balas Pesan

Gambar 83. Rancangan Antar Muka Baca Pesan (Staf)

m. Rancangan Antar Muka Menu *Users*

Menu *Users* digunakan oleh Administrator atau Staf untuk mengelola data pengguna. Rancangan antar muka menu *Users* untuk pengguna dengan hak akses Administrator disajikan pada Gambar 84 dan Gambar 85.

Bimbingan Skripsi Online Judul Upload Proposal List Proposal Dosen Hak Akses Pesan Users Profil Bimbingan logout

Sub menu
* Tambah data

Daftar Users

NIM/NIP	Nama	Alamat	Kontak	Opsi
				Detail/Ubah
				Detail/Ubah
				Detail/Ubah
				Detail/Ubah
				Detail/Ubah
				Detail/Ubah

Gambar 84. Rancangan Antar Muka Menu *Users* (Administrator)

Bimbingan Skripsi Online
[Judul](#)
[Dosen](#)
[Hak Akses](#)
[Pesan](#)
[Users](#)
[Profil](#)
[Logout](#)

Sub menu
* Tambah data

Daftar Users

NIM/NIP	Nama	Alamat	Kontak	Opsi
				Detail/Ubah
				Detail/Ubah
				Detail/Ubah
				Detail/Ubah
				Detail/Ubah
				Detail/Ubah
				Detail/Ubah

Gambar 85. Rancangan Antar Muka Menu *Users* (Staf)

Rancangan antar muka form Tambah *User* disajikan pada Gambar 86, dan Gambar 87.

Bimbingan Skripsi Online
[Judul](#)
[Upload Proposal](#)
[List Proposal](#)
[Dosen](#)
[Hak Akses](#)
[Pesan](#)
[Users](#)
[Profil](#)
[Bimbingan](#)
[logout](#)

Sub menu
* Lihat Daftar

Tambah User

NIM/NIP

Password

Nama

Alamat

Kontak

Gambar 86. Rancangan Antar Muka Form Tambah *User* (Administrator)

Bimbingan Skripsi Online		Judul	Dosen	Hak Akses	Pesan	Users	Profil	Logout
<div>Sub menu</div> <div>* Lihat Daftar</div>	<div>Tambah User</div> <div> <div>NIM/NIP</div> <div> <input type="text"/> </div> </div> <div> <div>Password</div> <div> <input type="password"/> </div> </div> <div> <div>Nama</div> <div> <input type="text"/> </div> </div> <div> <div>Alamat</div> <div> <input type="text"/> </div> </div> <div> <div>Kontak</div> <div> <input type="text"/> </div> </div> <div>Tambah</div>							

Gambar 87. Rancangan Antar Muka Form Tambah User (Staf)

Rancangan antar muka detail data *User* disajikan pada Gambar 88, dan Gambar 89.

Bimbingan Skripsi Online		Judul	Upload Proposal	List Proposal	Dosen	Hak Akses	Pesan	Users	Profil	Bimbingan	logout
<div>Sub menu</div> <div> * Tambah data * Ubah data * Lihat Daftar </div>	<div>Detail User</div> <div> <div>NIM/NIP</div> <div>:</div> </div> <div> <div>Nama</div> <div>:</div> </div> <div> <div>Alamat</div> <div>:</div> </div> <div> <div>Kontak</div> <div>:</div> </div>										

Gambar 88. Rancangan Antar Muka Detail User (Administrator)

Bimbingan Skripsi Online		Judul	Dosen	Hak Akses	Pesan	Users	Profil	Logout
<div>Sub menu</div> <ul style="list-style-type: none"> * Tambah data * Ubah data * Lihat Daftar 	<div>Detail User</div> <div> NIM/NIP : Nama : Alamat : Kontak : </div>							

Gambar 89. Rancangan Antar Muka Detail User (Staf)

n. Rancangan Antar Muka Menu Profil

Menu Profil digunakan pengguna untuk melihat data pribadinya. Menu ini akan langsung ditampilkan setelah *login* pengguna berhasil. Rancangan antar muka menu Profil disajikan pada Gambar 90, Gambar 91, Gambar 92, Gambar 93, dan Gambar 94.

Bimbingan Skripsi Online		Judul	Upload Proposal	List Proposal	Dosen	Hak Akses	Pesan	Users	Profil	Bimbingan	logout
<div>Sub menu</div> <ul style="list-style-type: none"> * Ubah data * Ganti Password 	<div>Profil</div> <div> NIM/NIP : Nama : Alamat : Kontak : </div>										

Gambar 90. Rancangan Antar Muka Profil (Administrator)

Bimbingan Skripsi Online		Judul	Dosen	Pesan	Profil	Bimbingan	Logout
<div>Sub menu</div> <div>* Ubah data</div> <div>* Ganti Password</div>		<div>Profil</div> <div> <div>NIM/NIP</div> <div>:</div> </div> <div> <div>Nama</div> <div>:</div> </div> <div> <div>Alamat</div> <div>:</div> </div> <div> <div>Kontak</div> <div>:</div> </div>					

Gambar 91. Rancangan Antar Muka Profil (Dosen)

Bimbingan Skripsi Online		Judul	List Proposal	Dosen	Pesan	Profil	Logout
<div>Sub menu</div> <div>* Ubah data</div> <div>* Ganti Password</div>		<div>Profil</div> <div> <div>NIM/NIP</div> <div>:</div> </div> <div> <div>Nama</div> <div>:</div> </div> <div> <div>Alamat</div> <div>:</div> </div> <div> <div>Kontak</div> <div>:</div> </div>					

Gambar 92. Rancangan Antar Muka Profil (Kaprod)

Bimbingan Skripsi Online		Judul	Upload Proposal	Dosen	Pesan	Profil	Logout
<div>Sub menu</div> <div>* Ubah data</div> <div>* Ganti Password</div>		<div>Profil</div> <div> <div>NIM/NIP</div> <div>:</div> </div> <div> <div>Nama</div> <div>:</div> </div> <div> <div>Alamat</div> <div>:</div> </div> <div> <div>Kontak</div> <div>:</div> </div>					

Gambar 93. Rancangan Antar Muka Profil (Mahasiswa)

Bimbingan Skripsi Online		Judul	Dosen	Hak Akses	Pesan	Users	Profil	Logout
<div>Sub menu</div> <div>* Ubah data</div> <div>* Ganti Password</div>		<div>Profil</div> <div> <div>NIM/NIP</div> <div>:</div> </div> <div> <div>Nama</div> <div>:</div> </div> <div> <div>Alamat</div> <div>:</div> </div> <div> <div>Kontak</div> <div>:</div> </div>						

Gambar 94. Rancangan Antar Muka Profil (Staf)

C. Implementasi Sistem

Implementasi sistem merupakan penerapan dari seluruh rancangan yang sudah dibuat sebelumnya dengan aplikasi program yang sudah ditentukan. Implementasi sistem bertujuan untuk mengkonfirmasi modul program rancangan

pada para pelaku sistem sehingga pengguna (*user*) dapat memberikan masukan kepada pengembangan sistem.

1. Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk mengimplementasikan sistem yang baru dibangun terdiri dari seperangkat komputer. Spesifikasi komputer yang digunakan menjalankan aplikasi yang dibangun adalah sebagai berikut:

- a. *Processor* Pentium IV 2,10 GHZ.
- b. *Hardisk* berkapasitas 249 GB.
- c. *Memory* berkapasitas 2.00 GB.

2. Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk mengimplementasikan sistem yang baru dibangun ini terdiri atas dua jenis yaitu perangkat lunak sistem operasi dan perangkat lunak pendukung. Perangkat lunak yang dibutuhkan adalah sebagai berikut:

- a. Sistem operasi Windows 7.
- b. *software Sublime 2.0* sebagai *text editor*.
- c. *Database Management System* (DBMS) MySQL Xampp 7.
- d. *Web browser Mozilla Firefox*.
- e. *Framework Yii*.

3. Implementasi Antar Muka

Berikut adalah contoh tampilan antar muka Aplikasi Bimbingan Skripsi *Online* yang telah dibuat:

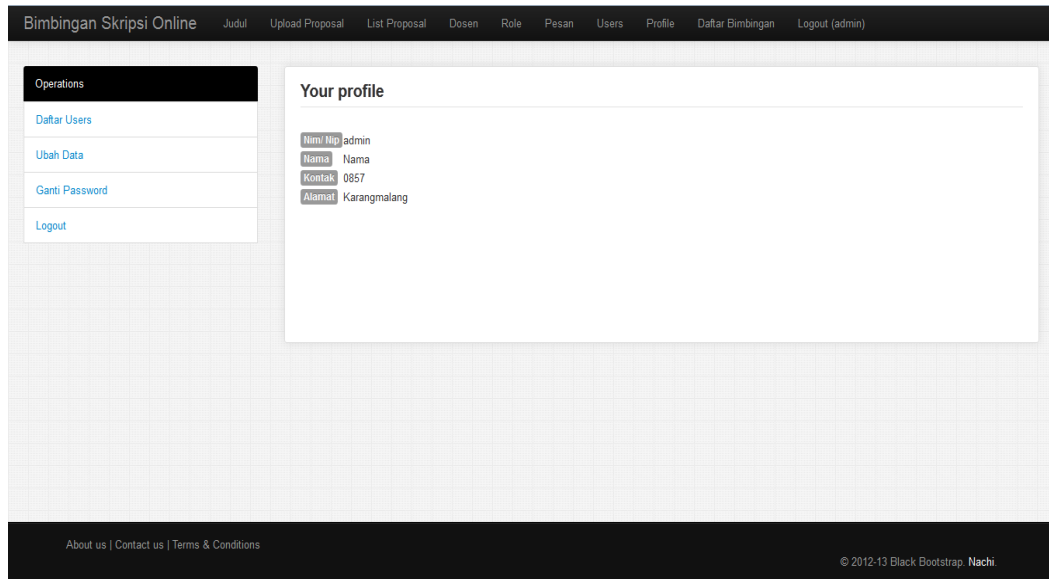
a. Tampilan Antar Muka Halaman *Login*

Halaman *login* adalah halaman pertama yang ditampilkan kepada pengguna yang belum *login* ketika mengakses aplikasi bimbingan skripsi *online*. Halaman *login* terdiri 2 *field* (kotak isian), yaitu NIM/NIP dan *password*. Tampilan antar muka halaman *login* disajikan pada Gambar 95. Kode program untuk Gambar 95 bisa dilihat pada lampiran halaman 153 dengan judul “login.php”.

Gambar 95. Tampilan Antar Muka Halaman Login

b. Tampilan Antar Muka Menu dan Profil Pengguna (Administrator)

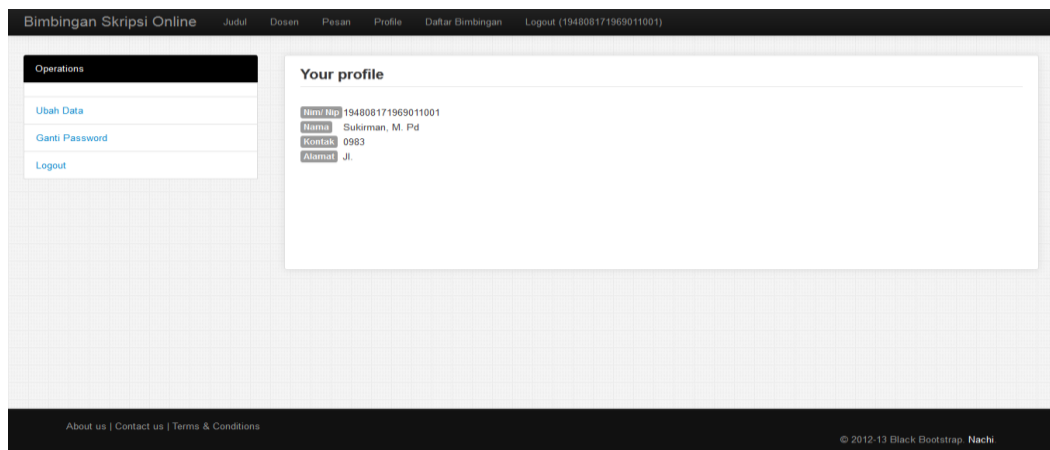
Pengguna dengan hak akses Administrator dapat mengakses sembilan menu setelah *login*, yaitu menu Judul, *Upload Proposal*, List Proposal, Dosen, Hak akses, Pesan, *Users*, Profil, dan *Logout*. Tampilan antar muka menu dan profil pengguna dengan hak akses Administrator disajikan pada Gambar 96. Kode program untuk Gambar 96 - 100 bisa dilihat pada lampiran halaman 265 dengan judul “profil.php”.



Gambar 96. Tampilan Antar Muka Menu Pengguna (Administrator)

c. Tampilan Antar Muka Menu dan Profil Pengguna (Dosen)

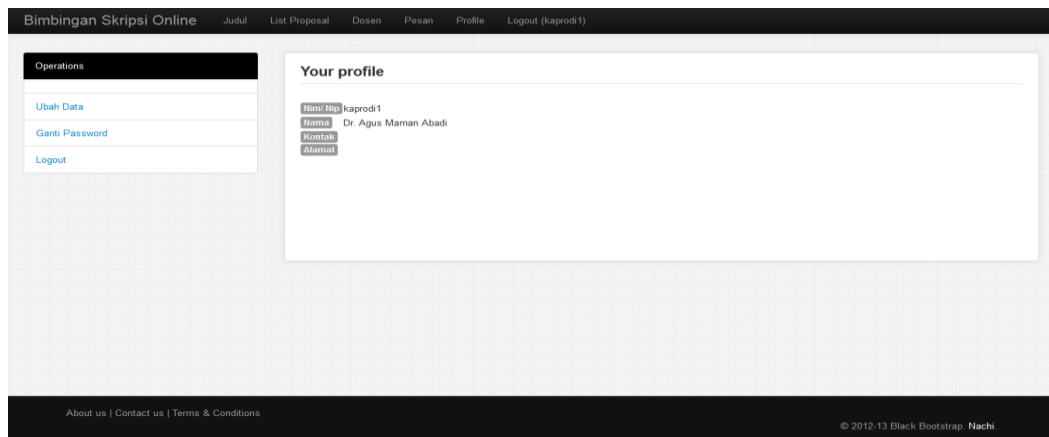
Pengguna dengan hak akses Dosen dapat mengakses lima menu setelah *login*, yaitu menu Judul, Dosen, Pesan, Profil, dan *Logout*. Tampilan antar muka menu dan profil pengguna dengan hak akses Dosen disajikan pada Gambar 97.



Gambar 97. Tampilan Antar Muka Menu Pengguna (Dosen)

d. Tampilan Antar Muka Menu dan Profil Pengguna (Kaprodi)

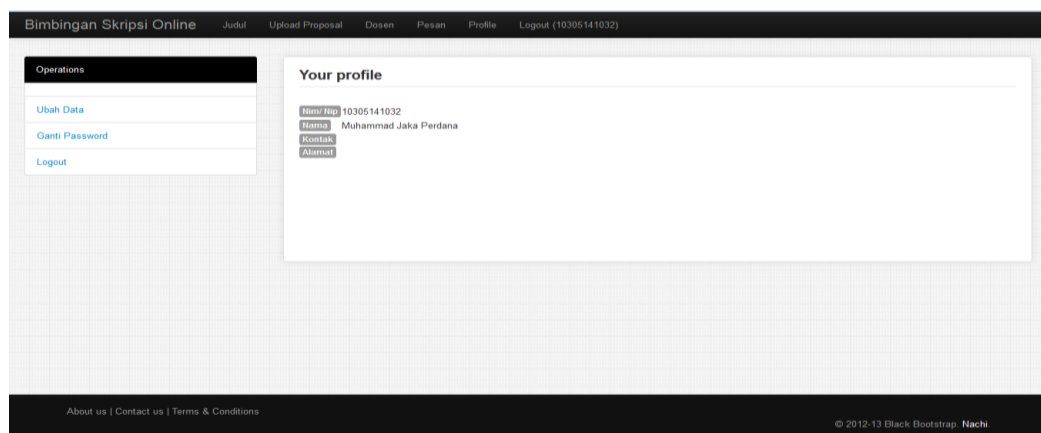
Pengguna dengan hak akses Kaprodi dapat mengakses enam menu setelah *login*, yaitu menu Judul, List Proposal, Dosen, Pesan, Profil, dan *Logout*. Tampilan antar muka pengguna dengan hak akses Kaprodi disajikan pada Gambar 98.



Gambar 98. Tampilan Antar Muka Menu Pengguna (Kaprodi)

e. Tampilan Antar Muka Menu dan Profil Pengguna (Mahasiswa)

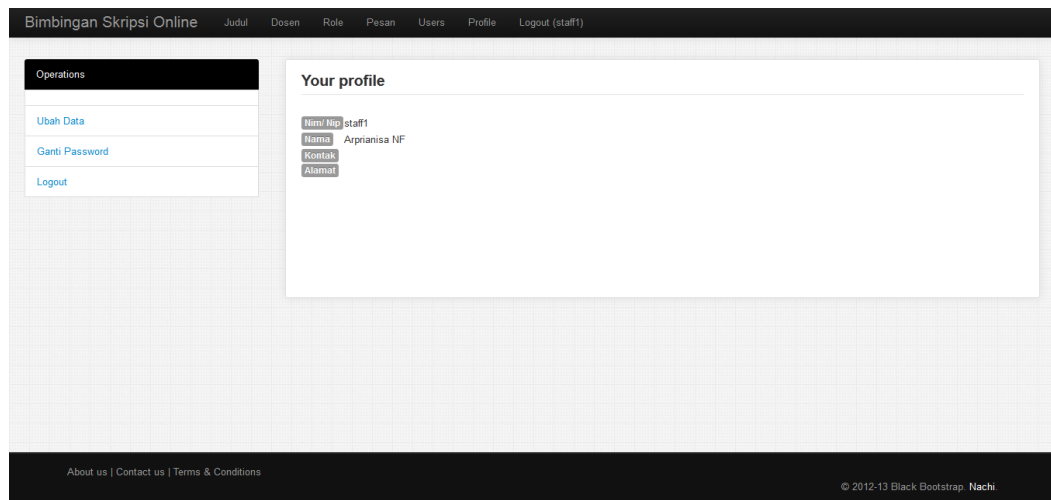
Pengguna dengan hak akses Mahasiswa dapat mengakses 6 menu setelah *login*, yaitu menu Judul, Upload Proposal, Dosen, Pesan, Profil, dan *Logout*. Tampilan antar muka pengguna dengan hak akses Mahasiswa disajikan pada Gambar 99.



Gambar 99. Tampilan Antar Muka Menu Pengguna (Mahasiswa)

f. Tampilan Antar Muka Menu dan Profil Pengguna (Staf)

Pengguna dengan hak akses Staf dapat mengakses tujuh menu setelah *login*, yaitu menu Judul, Dosen, Hak akses, Pesan, *Users*, Profil, dan *Logout*. Tampilan antar muka pengguna dengan hak akses Mahasiswa disajikan pada Gambar 100.



Gambar 100. Tampilan Antar Muka Menu Pengguna (Staf)

g. Tampilan Antar Muka Menu Judul

Menu Judul digunakan untuk mengelola dan menampilkan data mahasiswa Jurdikmat UNY yang telah mengajukan judul skripsi dan telah memulai bimbingan skripsi. Data yang ditampilkan adalah NIM, Nama Mahasiswa, Judul Skripsi, Tanggal Masuk, Tanggal Ujian, dan Detail. Tampilan antar muka menu Judul disajikan pada Gambar 101, Gambar 102, Gambar 103, Gambar 104, dan Gambar 105. Kode program untuk Gambar 101 - 105 bisa dilihat pada lampiran halaman 166 dengan judul “profil.php”.

Bimbingan Skripsi Online									
Judul Upload Proposal List Proposal Dosen Hak Akses Pesan Users Profile Daftar Bimbingan Logout (admin)									
Operations									
Tambah Daftar									
Daftar Skripsi Mahasiswa									
Totally 60 results have been found									
Nim_mhs	Nama_mhs	Judul_skripsi	Tgl_masuk	Tgl_ujian	Actions				
10305141032	Muhammad Jaka P.	Aplikasi Bimbingan Skripsi Online Berbasis Web	2014-03-24	2014-07-14	Q				
10305141031	Ericolion Maubrani	Sistem Hasil Kali Pada Bidang	2014-03-20	0000-00-00	Q				
10305141019	Rizal Arrosyid	Pengendalian Gerak Satelit Dean Metode Linear Quadratic Regulator (LQR)	2014-02-18	2014-06-12	Q				
10305144027	Mita Riana	Model Antrian Untuk Waktu Tunggu Kendaraan Di Persimpangan Lalu Lintas	2014-03-13	0000-00-00	Q				
10305141008	Fajarani Juliaristi	Peramalan Kasus Demam Berdarah di D.I. Yogyakarta dengan Metode Radial Basis Function Neural Network (RBFNN)	2014-02-20	0000-00-00	Q				
10305144042	Ambar Sito Jati	Ruang Metrik Fuzzy dan Sifat-sifatnya	2014-01-20	0000-00-00	Q				
10305141016	Arif Munandar	Aplikasi Metode Fuzzy-Two Step Filter Untuk Mereduksi Impulse Noise Pada Citra Bertipe RGB	2014-01-13	2014-04-16	Q				

Gambar 101. Tampilan Antar Muka Menu Judul (Administrator)

Bimbingan Skripsi Online									
Judul Dosen Pesan Profile Daftar Bimbingan Logout (194908171969011001)									
Daftar Skripsi Mahasiswa									
Totally 60 results have been found									
Nim_mhs	Nama_mhs	Judul_skripsi	Tgl_masuk	Tgl_ujian	Actions				
10305141032	Muhammad Jaka P.	Aplikasi Bimbingan Skripsi Online Berbasis Web	2014-03-24	2014-07-14	Q				
10305141031	Ericolion Maubrani	Sistem Hasil Kali Pada Bidang	2014-03-20	0000-00-00	Q				
10305141019	Rizal Arrosyid	Pengendalian Gerak Satelit Dean Metode Linear Quadratic Regulator (LQR)	2014-02-18	2014-06-12	Q				
10305144027	Mita Riana	Model Antrian Untuk Waktu Tunggu Kendaraan Di Persimpangan Lalu Lintas	2014-03-13	0000-00-00	Q				
10305141008	Fajarani Juliaristi	Peramalan Kasus Demam Berdarah di D.I. Yogyakarta dengan Metode Radial Basis Function Neural Network (RBFNN)	2014-02-20	0000-00-00	Q				
10305144042	Ambar Sito Jati	Ruang Metrik Fuzzy dan Sifat-sifatnya	2014-01-20	0000-00-00	Q				
10305141016	Arif Munandar	Aplikasi Metode Fuzzy-Two Step Filter Untuk Mereduksi Impulse Noise Pada Citra Bertipe RGB	2014-01-13	2014-04-16	Q				
10305141022	Lidyana Candrawati	Model Penyebaran Penyakit Hepatitis C Melalui Jarum Suntik Oleh Pengguna Narkoba	2013-12-02	2014-04-08	Q				
10305141027	Rian Anggara Putra	Efektivitas Metode Sequential Insertion dan Metode Nearest Insertion dalam Penentuan Rute Kendaraan Pengangkutan Sampah di Kota Yogyakarta	2014-03-07	2014-07-08	Q				
10305144006	Arif Kurniawan	Mengukur Tingkat Kesiapan Penerapan E-learning di Sekolah Menengah Atas	2014-02-27	2014-06-19	Q				

Gambar 102. Tampilan Antar Muka Menu Judul (Dosen)

Bimbingan Skripsi Online					
	Judul	List Proposal	Dosen	Pesan	Profile Logout (kaprodi1)

Operations	Daftar Skripsi Mahasiswa				
Tambah Daftar	Totaly 60 results have been found				
Nim_mhs	Nama_mhs	Judul_skripsi	Tgl_masuk	Tgl_ujian	Actions
10305141032	Muhammad Jaka P.	Aplikasi Bimbingan Skripsi Online Berbasis Web	2014-03-24	2014-07-14	Q E
10305141031	Ericolion Maubrani	Sistem Hasil Kali Pada Bidang	2014-03-20	0000-00-00	Q E
10305141019	Rizal Arrosyid	Pengendalian Gerak Satelit Dean Metode Linear Quadratic Regulator (LQR)	2014-02-18	2014-06-12	Q E
10305144027	Mita Riana	Model Antrian Untuk Waktu Tunggu Kendaraan Di Persimpangan Lalu Lintas	2014-03-13	0000-00-00	Q E
10305141008	Fajarani Juliaristi	Peramalan Kasus Demam Berdarah di D.I. Yogyakarta dengan Metode Radial Basis Function Neural Network (RBFNN)	2014-02-20	0000-00-00	Q E
10305144042	Ambar Sito Jati	Ruang Metrik Fuzzy dan Sifat-sifatnya	2014-01-20	0000-00-00	Q E
10305141016	Arif Munandar	Aplikasi Metode Fuzzy-Two Step Filter Untuk Mereduksi Impulse Noise Pada Citra Bertipe RGB	2014-01-13	2014-04-16	Q E

Gambar 103. Tampilan Antar Muka Menu Judul (Kaprodi)

Bimbingan Skripsi Online					
	Judul	Upload Proposal	Dosen	Pesan	Profile Logout (10305141032)

	Daftar Skripsi Mahasiswa				
	Totaly 60 results have been found				
Nim_mhs	Nama_mhs	Judul_skripsi	Tgl_masuk	Tgl_ujian	Actions
10305141032	Muhammad Jaka P.	Aplikasi Bimbingan Skripsi Online Berbasis Web	2014-03-24	2014-07-14	Q
10305141031	Ericolion Maubrani	Sistem Hasil Kali Pada Bidang	2014-03-20	0000-00-00	Q
10305141019	Rizal Arrosyid	Pengendalian Gerak Satelit Dean Metode Linear Quadratic Regulator (LQR)	2014-02-18	2014-06-12	Q
10305144027	Mita Riana	Model Antrian Untuk Waktu Tunggu Kendaraan Di Persimpangan Lalu Lintas	2014-03-13	0000-00-00	Q
10305141008	Fajarani Juliaristi	Peramalan Kasus Demam Berdarah di D.I. Yogyakarta dengan Metode Radial Basis Function Neural Network (RBFNN)	2014-02-20	0000-00-00	Q
10305144042	Ambar Sito Jati	Ruang Metrik Fuzzy dan Sifat-sifatnya	2014-01-20	0000-00-00	Q
10305141016	Arif Munandar	Aplikasi Metode Fuzzy-Two Step Filter Untuk Mereduksi Impulse Noise Pada Citra Bertipe RGB	2014-01-13	2014-04-16	Q
10305141022	Lidyana Candrawati	Model Penyebaran Penyakit Hepatitis C Melalui Jarum Suntik Oleh Pengguna Narkoba	2013-12-02	2014-04-08	Q
10305141027	Rian Anggara Putra	Efektivitas Metode Sequential Insertion dan Metode Nearest Insertion dalam Penentuan Rute Kendaraan Pengangkut Sampah di Kota Yogyakarta	2014-03-07	2014-07-08	Q
10305144006	Arif Kurniawan	Mengukur Tingkat Kesiapan Penerapan E-learning di Sekolah Menengah Atas	2014-02-27	2014-06-19	Q

Gambar 104. Tampilan Antar Muka Menu Judul (Mahasiswa)

Bimbingan Skripsi Online						
Judul	Dosen	Hak Akses	Pesan	Users	Profile	Logout (staff1)

Operations						
Tambah Daftar						

Daftar Skripsi Mahasiswa						
Totally 60 results have been found						
Nim_mhs	Nama_mhs	Judul_skripsi	Tgl_masuk	Tgl_ujian	Actions	
10305141032	Muhammad Jaka P.	Aplikasi Bimbingan Skripsi Online Berbasis Web	2014-03-24	2014-07-14	Q	E
10305141031	Ericolion Maubrani	Sistem Hasil Kali Pada Bidang	2014-03-20	0000-00-00	Q	E
10305141019	Rizal Arrosyid	Pengendalian Gerak Satelit Dean Metode Linear Quadratic Regulator (LQR)	2014-02-18	2014-06-12	Q	E
10305144027	Mita Riana	Model Antrian Untuk Waktu Tunggu Kendaraan Di Persimpangan Lalu Lintas	2014-03-13	0000-00-00	Q	E
10305141008	Fajarani Juliaristi	Peramalan Kasus Demam Berdarah di D.I. Yogyakarta dengan Metode Radial Basis Function Neural Network (RBFNN)	2014-02-20	0000-00-00	Q	E
10305144042	Ambar Sito Jati	Ruang Metrik Fuzzy dan Sifat-sifatnya	2014-01-20	0000-00-00	Q	E
10305141016	Arif Munandar	Aplikasi Metode Fuzzy-Two Step Filter Untuk Mereduksi Impulse Noise Pada Citra Ber tipe RGB	2014-01-13	2014-04-16	Q	E

Gambar 105. Tampilan Antar Muka Menu Judul (Staf)

Tampilan menu tambah judul skripsi hanya bisa diakses oleh pengguna dengan hak akses Administrator, dan Staf. Menu ini terdiri dari 9 *field*, yaitu NIM, Nama Mahasiswa, Fokus Studi, Judul Skripsi, Pembimbing, Tanggal Masuk, Tanggal Ujian, Ketua Penguji, Sekretaris Penguji, Penguji Utama, dan Penguji Pendamping. Tampilan menu tambah judul disajikan pada Gambar 106, dan Gambar 107. Kode program untuk Gambar 106 – 107 bisa dilihat pada lampiran halaman 167 dengan judul “create.php”.

Bimbingan Skripsi Online											
Judul	Upload Proposal	List Proposal	Dosen	Hak Akses	Pesan	Users	Profile	Daftar Bimbingan	Logout (admin)		

Operations											
Daftar Skripsi											

Tambah Judul Skripsi											
Fields dengan tanda * harus diisi.											
NIM *											
Nama Mahasiswa *											
Fokus Studi * Aljabar											
Judul Skripsi *											
Pembimbing * ---Dosen---											
Tanggal Masuk *											
Tanggal Ujian *											
Ketua Penguji ---Dosen---											
Sekretaris Penguji ---Dosen---											
Penguji Utama ---Dosen---											
Penguji Pendamping ---Dosen---											
Tambah											

About us Contact us Terms & Conditions						© 2012-13 Black Bootstrap. Nach!					
--	--	--	--	--	--	----------------------------------	--	--	--	--	--

Gambar 106. Tampilan Antar Muka Form Tambah Data Judul (Administrator)

Bimbingan Skripsi Online Judul Dosen Hak Akses Pesan Users Profile Logout (staff1)

Operations

[Daftar Skripsi](#)

Tambah Judul Skripsi

Fields dengan tanda * harus diisi.

NIM *

Nama Mahasiswa *

Fokus Studi * Aljabar

Judul Skripsi *

Pembimbing * ---Dosen---

Tanggal Masuk *

Tanggal Ujian *

Ketua Penguji ---Dosen---

Sekretaris Penguji ---Dosen---

Penguji Utama ---Dosen---

Penguji Pendamping ---Dosen---

Tambah

Gambar 107. Tampilan Antar Muka Form Tambah Data Judul (Staf)

Tampilan detail judul disajikan pada Gambar 108, Gambar 109, Gambar 110, Gambar 111, dan Gambar 112, menampilkan informasi lengkap dari judul skripsi yang terdiri dari 9 *field* yang sesuai dengan menu tambah data. Kode program untuk Gambar 108 - 112 bisa dilihat pada lampiran halaman 169 dengan judul “view.php”.

Bimbingan Skripsi Online Judul Upload Proposal List Proposal Dosen Hak Akses Pesan Users Profile Daftar Bimbingan Logout (admin)

Operations

[Tambah Daftar](#)

[Update Data](#)

[Hapus Data](#)

[Daftar Judul Skripsi](#)

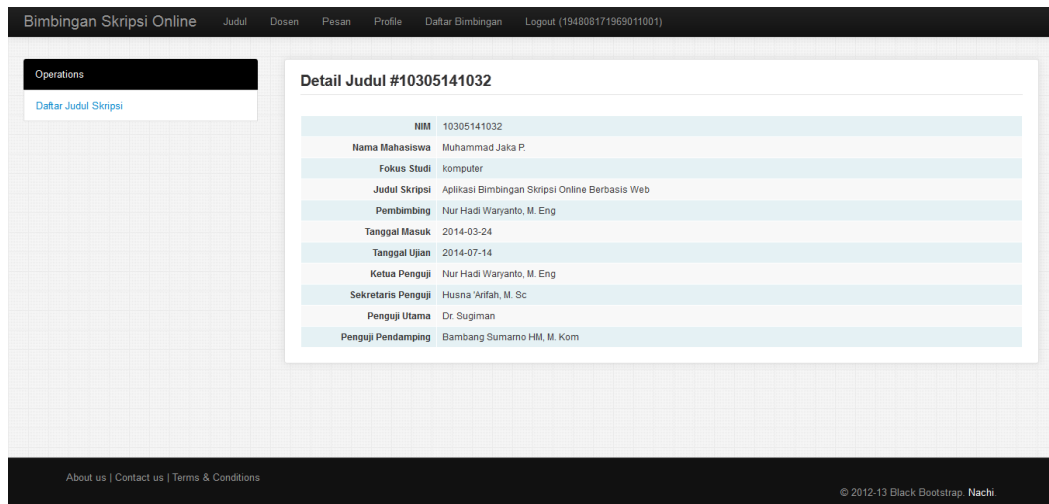
Detail Judul #10305141032

NIM	10305141032
Nama Mahasiswa	Muhammad Jaka P.
Fokus Studi	komputer
Judul Skripsi	Aplikasi Bimbingan Skripsi Online Berbasis Web
Pembimbing	Nur Hadi Waryanto, M. Eng
Tanggal Masuk	2014-03-24
Tanggal Ujian	2014-07-14
Ketua Penguji	Nur Hadi Waryanto, M. Eng
Sekretaris Penguji	Husna 'Arifah, M. Sc
Penguji Utama	Dr. Sugiman
Penguji Pendamping	Bambang Sumarno HM, M. Kom

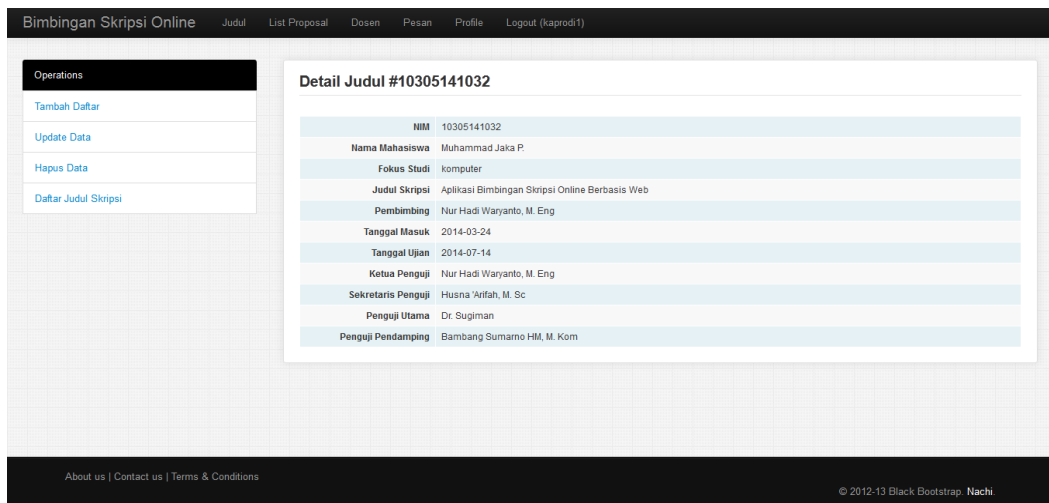
About us | Contact us | Terms & Conditions

© 2012-13 Black Bootstrap. Nachi

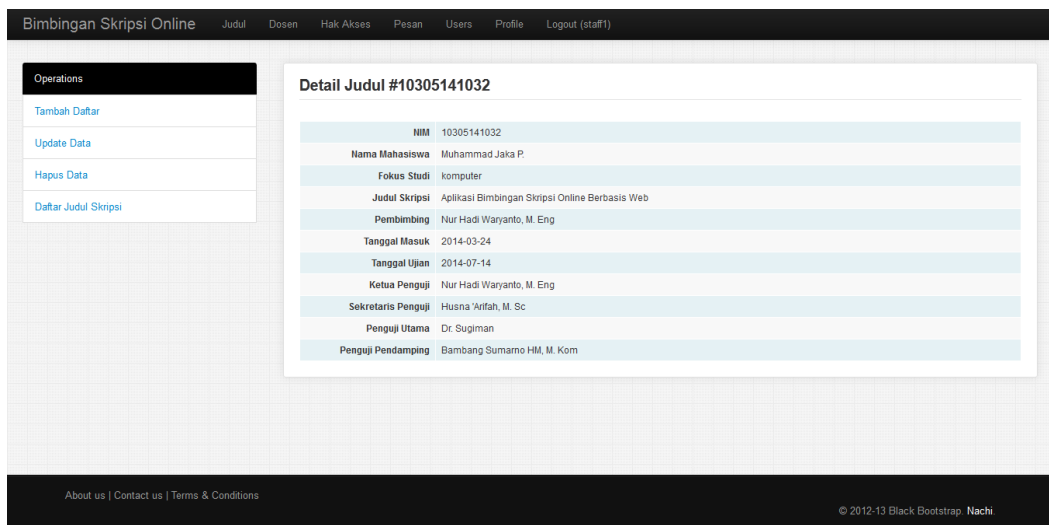
Gambar 108. Tampilan Antar Muka Detail Data Judul (Administrator)



Gambar 109. Tampilan Antar Muka Detail Data Judul (Dosen)



Gambar 110. Tampilan Antar Muka Detail Data Judul (Kaprodi)



Gambar 111. Tampilan Antar Muka Detail Data Judul (Staf)

h. Tampilan Antar Muka Menu *Upload Proposal*

Menu *Upload Proposal* digunakan oleh pengguna dengan hak akses Mahasiswa untuk mengajukan proposal skripsi kepada Kaprodi dengan meng-*upload* file proposal skripsinya. Tampilan antar muka menu *Upload Proposal* disajikan pada Gambar 112, dan Gambar 113. Kode program untuk Gambar 112 - 113 bisa dilihat pada lampiran halaman 178 dengan judul “create.php”.

The screenshot shows a web application interface for 'Bimbingan Skripsi Online'. The top navigation bar includes links: Bimbingan Skripsi Online, Judul, Upload Proposal, List Proposal, Dosen, Hak Akses, Pesan, Users, Profile, Daftar Bimbingan, and Logout (admin). The main content area features a form titled 'Upload Proposal' with the instruction 'Fields dengan tanda * harus diisi.' Below this, there is a text input field for 'Judul Proposal *'. Underneath, there is a 'File Proposal' label, a 'Browse...' button, and the text 'No file selected.'. At the bottom of the form is an 'Upload' button. The footer contains links 'About us | Contact us | Terms & Conditions' and a copyright notice '© 2012-13 Black Bootstrap. Nachi.'

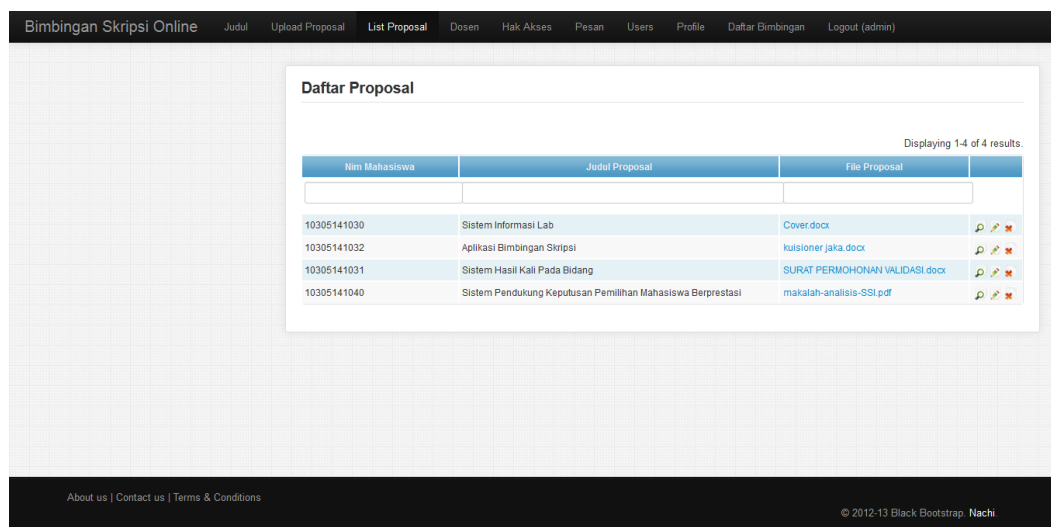
Gambar 112. Tampilan Antar Muka Menu Upload Proposal (Administrator)

The screenshot shows the same 'Upload Proposal' form as in Gambar 112, but for a Mahasiswa user. The top navigation bar includes links: Bimbingan Skripsi Online, Judul, Upload Proposal, Dosen, Pesan, Profile, and Logout (10305141032). The form structure is identical, with the 'Judul Proposal *' field, 'File Proposal' section with 'Browse...' button and 'No file selected.' text, and an 'Upload' button. The footer is the same as in Gambar 112.

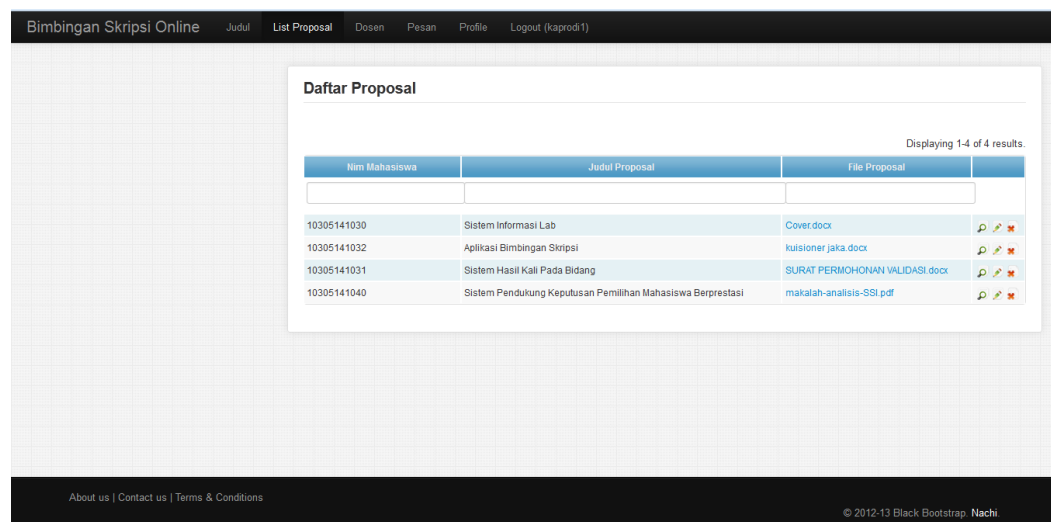
Gambar 113. Tampilan Antar Muka Menu Upload Proposal (Mahasiswa)

i. Tampilan Antar Muka Menu List Proposal

Menu List Proposal digunakan oleh pengguna dengan hak akses Kaprodi untuk *men-download file* proposal skripsi yang diajukan (*di-upload*) mahasiswa. Tampilan menu List Proposal disajikan pada Gambar 114, dan Gambar 115. Kode program untuk Gambar 114 - 115 bisa dilihat pada lampiran halaman 177 dengan judul “admin.php”.



Gambar 114. Tampilan Antar Muka Menu List Proposal (Administrator)



Gambar 115. Tampilan Antar Muka Menu List Proposal (Kaprodi)

j. Tampilan Antar Muka Menu Dosen

Menu Dosen digunakan untuk melihat dan mengelola daftar dosen Jurdikmat UNY. Data yang ditampilkan adalah data Nama Dosen, NIP, Tanggal Lahir, Bidang Keahlian, Jabatan, Gelar, dan Detail. Tampilan antar muka menu dosen disajikan pada Gambar 116, Gambar 117, Gambar 118, Gambar 119, dan Gambar 120. Kode program untuk Gambar 116 – 120 bisa dilihat pada lampiran halaman 187 dengan judul “admin.php”.

Nama_dosen	Nip	Bidang_keahlian	Jabatan	Pangkat	Actions
Sukirman, M. Pd.	194808171969011001	Pendidikan Ilmu Pasti, Pendidikan Matematika	Lektor Kepala (700)	Pembina Utama Muda / IVc	[Search] [Delete]
Sugiyono, M. Pd	195308251979031004	Pendidikan Ilmu Pasti, Pendidikan Matematika	Lektor Kepala (550)	Pembina TK I / IVb	[Search] [Delete]
Kuswari Hemawati, M. Kom	197604142005012002	Matematika, Ilmu Komputer	Lektor (300)	Penata TK I / III d	[Search] [Delete]
Bambang Sumarno HM, M. Kom	196802101988121001	Ilmu Komputer, Ilmu Komputer	Asisten Ahli	Penata Muda / IIIa	[Search] [Delete]
Kismiantini, M. Si	19790816200112001	Matematika, Statistika	Lektor (200)	Penata / IIIa	[Search] [Delete]
Prof. Marsigit	195707191983031004	Pendidikan Matematika, Primary Education, Filsafat Pendidikan Matematika	Guru Besar (850)	Pembina / IVa	[Search] [Delete]

Gambar 116. Tampilan Antar Muka Daftar Dosen (Administrator)

Nama_dosen	Nip	Bidang_keahlian	Jabatan	Pangkat	Actions
Sukirman, M. Pd.	194808171969011001	Pendidikan Ilmu Pasti, Pendidikan Matematika	Lektor Kepala (700)	Pembina Utama Muda / IVc	[Search]
Sugiyono, M. Pd	195308251979031004	Pendidikan Ilmu Pasti, Pendidikan Matematika	Lektor Kepala (550)	Pembina TK I / IVb	[Search]
Kuswari Hemawati, M. Kom	197604142005012002	Matematika, Ilmu Komputer	Lektor (300)	Penata TK I / III d	[Search]
Bambang Sumarno HM, M. Kom	196802101988121001	Ilmu Komputer, Ilmu Komputer	Asisten Ahli	Penata Muda / IIIa	[Search]
Kismiantini, M. Si	19790816200112001	Matematika, Statistika	Lektor (200)	Penata / IIIa	[Search]
Prof. Marsigit	195707191983031004	Pendidikan Matematika, Primary Education, Filsafat Pendidikan Matematika	Guru Besar (850)	Pembina / IVa	[Search]
Prof. Rusgianto Heri Santosa	194904171973031001	Pendidikan Ilmu Pasti, Teknologi Pendidikan, Teknologi Pendidikan	Guru Besar (850)	Pembina Utama Madya / IVd	[Search]
Husna 'Arifah, M. Sc	197810152002122001	Pendidikan Ilmu Pasti, Pendidikan Matematika	Asisten Ahli	III/b	[Search]

Gambar 117. Tampilan Antar Muka Daftar Dosen (Dosen)

Bimbingan Skripsi Online						
	Judul	List Proposal	Dosen	Pesan	Profile	Logout (kaprodi1)

Daftar Dosen						
Totally 15 results have been found						
Nama_dosen	Nip	Bidang_keahlian	Jabatan	Pangkat	Actions	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Sukirman, M. Pd.	194808171969011001	Pendidikan Ilmu Pasti, Pendidikan Matematika	Lektor Kepala (700)	Pembina Utama Muda / IVc	<input type="button" value="Q"/>	
Sugiyono, M. Pd	195308251979031004	Pendidikan Ilmu Pasti, Pendidikan Matematika	Lektor Kepala (550)	Pembina TK I / IVb	<input type="button" value="Q"/>	
Kuswari Hemawati, M. Kom	197604142005012002	Matematika, Ilmu Komputer	Lektor (300)	Penata TK I / IIIId	<input type="button" value="Q"/>	
Bambang Sumarmo HM, M. Kom	196802101988121001	Ilmu Komputer, Ilmu Komputer	Asisten Ahli	Penata Muda / IIIa	<input type="button" value="Q"/>	
Kismiantini, M. Si	19790816200112001	Matematika, Statistika	Lektor (200)	Penata / IIIa	<input type="button" value="Q"/>	
Prof. Marsigit	195707191983031004	Pendidikan Matematika, Primary Education, Filsafat Pendidikan Matematika	Guru Besar (850)	Pembina / IVa	<input type="button" value="Q"/>	
Prof. Ruscianto Heri Santosa	194904171973031001	Pendidikan Ilmu Pasti, Teknologi Pendidikan, Teknologi Pendidikan	Guru Besar (850)	Pembina Utama Madya / IVd	<input type="button" value="Q"/>	
Husna 'Anifah, M. Sc	197810152002122001	Pendidikan Ilmu Pasti, Pendidikan Matematika	Asisten Ahli	III/b	<input type="button" value="Q"/>	

localhost/contoh/index.php?r=dosen/admin

Gambar 118. Tampilan Antar Muka Daftar Dosen (Kaprodi)

Bimbingan Skripsi Online						
	Judul	Upload Proposal	Dosen	Pesan	Profile	Logout (10305141032)

Daftar Dosen						
Totally 15 results have been found						
Nama_dosen	Nip	Bidang_keahlian	Jabatan	Pangkat	Actions	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Sukirman, M. Pd.	194808171969011001	Pendidikan Ilmu Pasti, Pendidikan Matematika	Lektor Kepala (700)	Pembina Utama Muda / IVc	<input type="button" value="Q"/>	
Sugiyono, M. Pd	195308251979031004	Pendidikan Ilmu Pasti, Pendidikan Matematika	Lektor Kepala (550)	Pembina TK I / IVb	<input type="button" value="Q"/>	
Kuswari Hemawati, M. Kom	197604142005012002	Matematika, Ilmu Komputer	Lektor (300)	Penata TK I / IIIId	<input type="button" value="Q"/>	
Bambang Sumarmo HM, M. Kom	196802101988121001	Ilmu Komputer, Ilmu Komputer	Asisten Ahli	Penata Muda / IIIa	<input type="button" value="Q"/>	
Kismiantini, M. Si	19790816200112001	Matematika, Statistika	Lektor (200)	Penata / IIIa	<input type="button" value="Q"/>	
Prof. Marsigit	195707191983031004	Pendidikan Matematika, Primary Education, Filsafat Pendidikan Matematika	Guru Besar (850)	Pembina / IVa	<input type="button" value="Q"/>	
Prof. Ruscianto Heri Santosa	194904171973031001	Pendidikan Ilmu Pasti, Teknologi Pendidikan, Teknologi Pendidikan	Guru Besar (850)	Pembina Utama Madya / IVd	<input type="button" value="Q"/>	
Husna 'Anifah, M. Sc	197810152002122001	Pendidikan Ilmu Pasti, Pendidikan Matematika	Asisten Ahli	III/b	<input type="button" value="Q"/>	

Gambar 119. Tampilan Antar Muka Daftar Dosen (Mahasiswa)

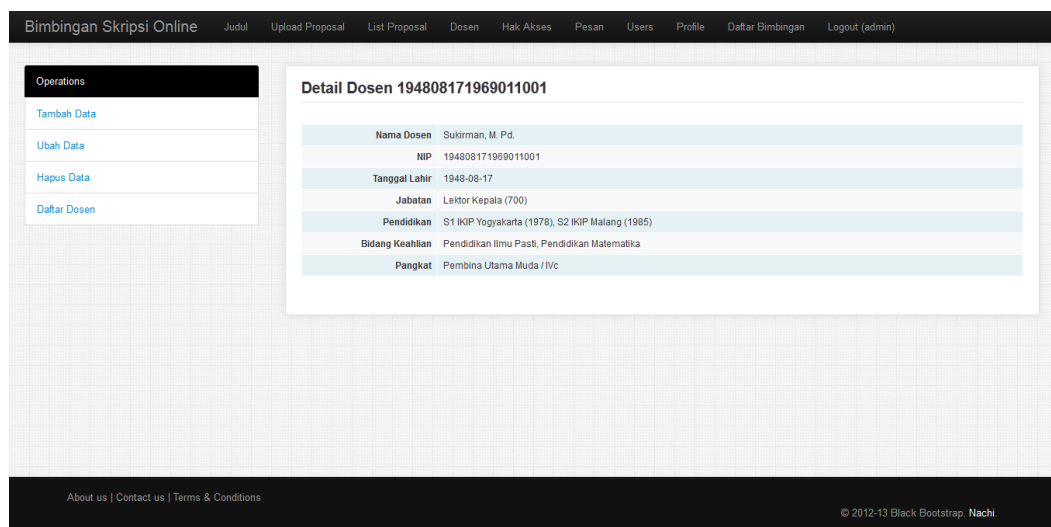
Bimbingan Skripsi Online						
	Judul	Dosen	Hak Akses	Pesan	Users	Profile
						Logout (staff1)

Daftar Dosen						
Totally 15 results have been found						
Nama_dosen	Nip	Bidang_keahlian	Jabatan	Pangkat	Actions	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Sukirman, M. Pd.	194808171969011001	Pendidikan Ilmu Pasti, Pendidikan Matematika	Lektor Kepala (700)	Pembina Utama Muda / IVc	<input type="button" value="Q"/> <input type="button" value="E"/>	
Sugiyono, M. Pd	195308251979031004	Pendidikan Ilmu Pasti, Pendidikan Matematika	Lektor Kepala (550)	Pembina TK I / IVb	<input type="button" value="Q"/> <input type="button" value="E"/>	
Kuswari Hemawati, M. Kom	197604142005012002	Matematika, Ilmu Komputer	Lektor (300)	Penata TK I / IIIId	<input type="button" value="Q"/> <input type="button" value="E"/>	
Bambang Sumarmo HM, M. Kom	196802101988121001	Ilmu Komputer, Ilmu Komputer	Asisten Ahli	Penata Muda / IIIa	<input type="button" value="Q"/> <input type="button" value="E"/>	
Kismiantini, M. Si	19790816200112001	Matematika, Statistika	Lektor (200)	Penata / IIIa	<input type="button" value="Q"/> <input type="button" value="E"/>	
Prof. Marsigit	195707191983031004	Pendidikan Matematika, Primary Education, Filsafat Pendidikan Matematika	Guru Besar (850)	Pembina / IVa	<input type="button" value="Q"/> <input type="button" value="E"/>	

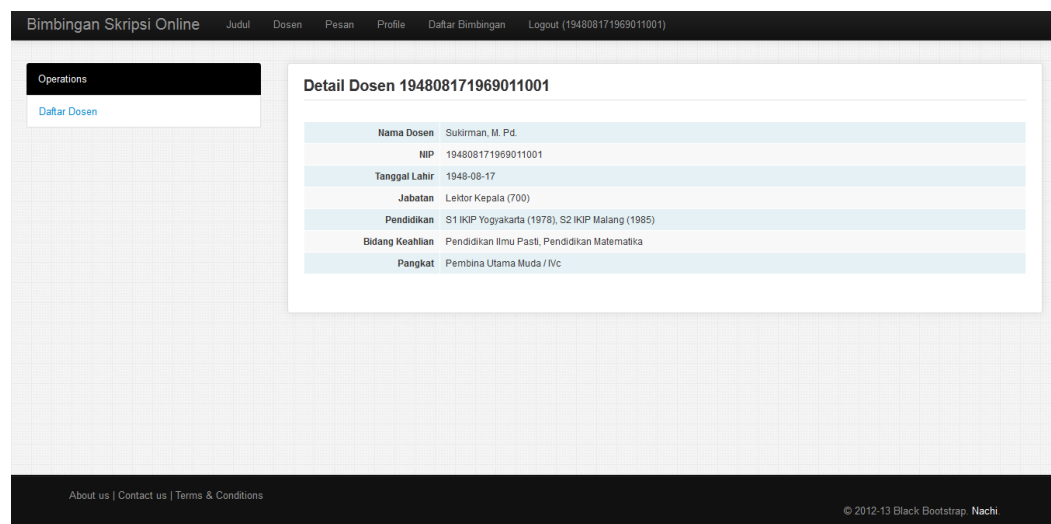
localhost/contoh/index.php?r=dosen/admin

Gambar 120. Tampilan Antar Muka Daftar Dosen (Staf)

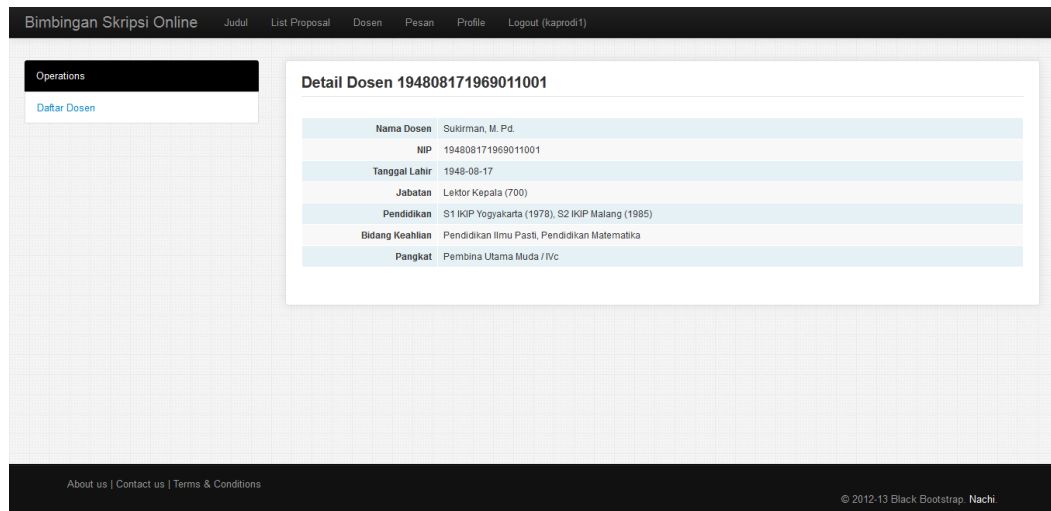
Tampilan detail data dosen disajikan pada Gambar 121, Gambar 122, Gambar 123, Gambar 124, dan Gambar 125, menampilkan data lengkap dari dosen yang sesuai dengan 7 *field* yang diisi di menu tambah data dosen. Kode program untuk Gambar 121 - 125 bisa dilihat pada lampiran halaman 190 dengan judul “view.php”.



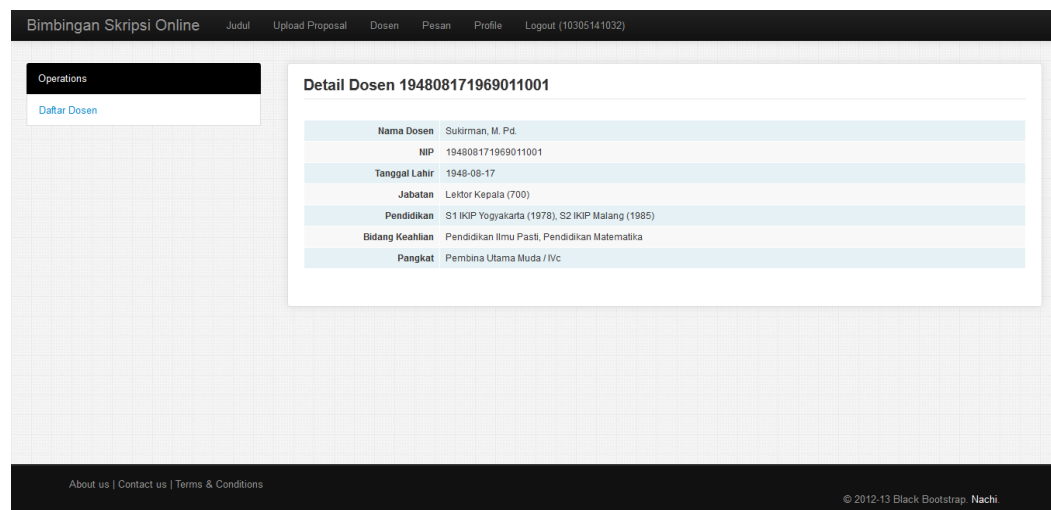
Gambar 121. Tampilan Antar Detail Data Dosen (Administrator)



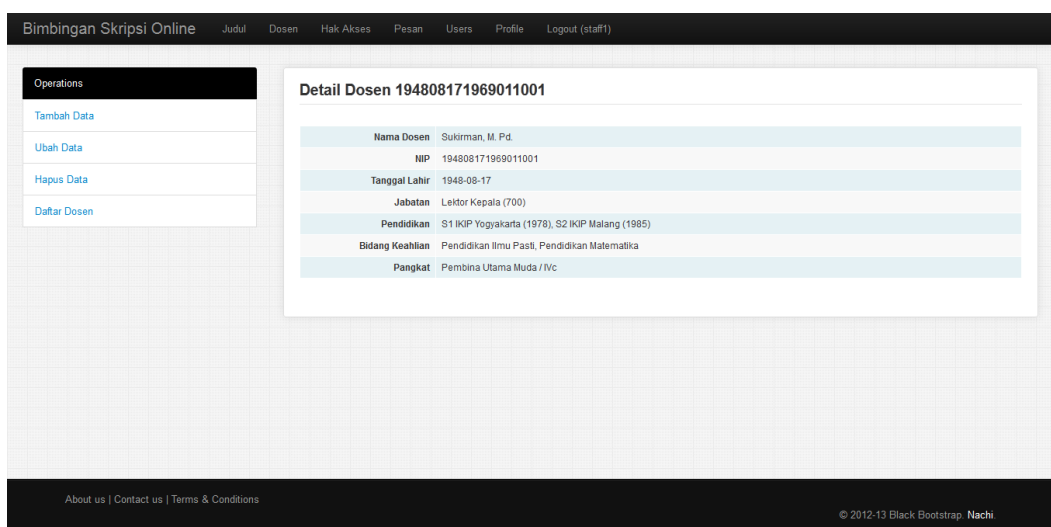
Gambar 122. Tampilan Antar Detail Data Dosen (Dosen)



Gambar 123. Tampilan Antar Detail Data Dosen (Kaprodi)



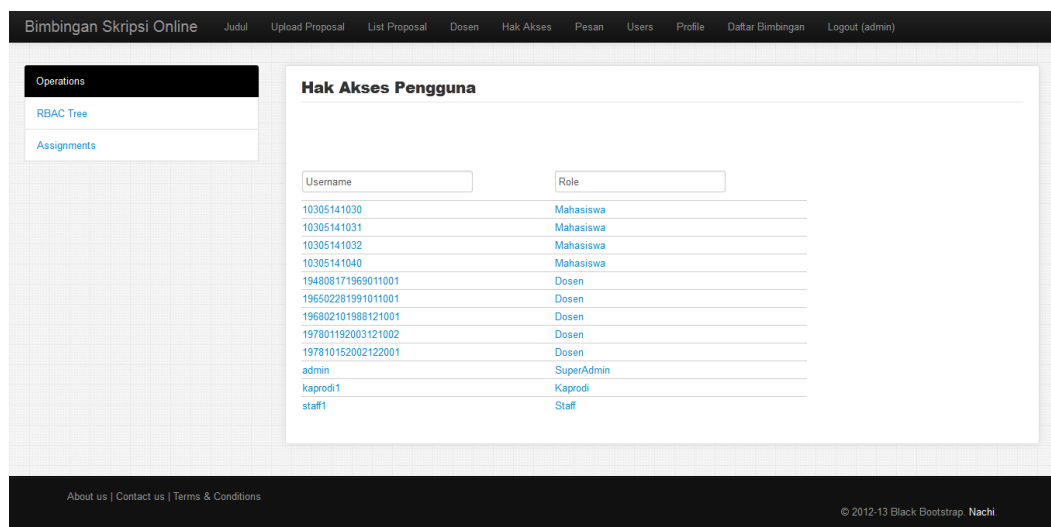
Gambar 124. Tampilan Antar Detail Data Dosen (Mahasiswa)



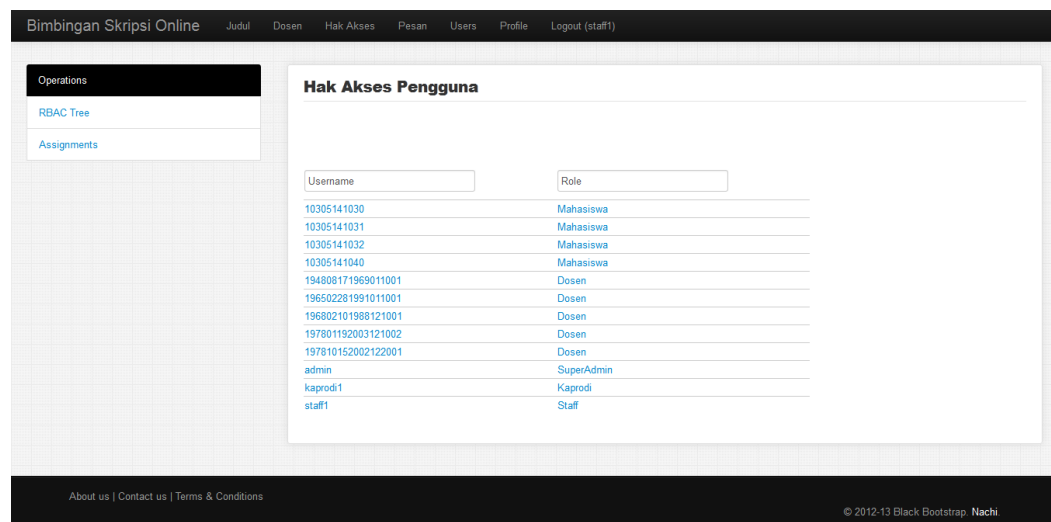
Gambar 125. Tampilan Antar Detail Data Dosen (Staf)

k. Tampilan Antar Muka Menu Hak Akses

Menu Hak akses digunakan oleh Administrator atau Staf untuk mengelola hak akses pengguna. Pengguna yang baru membuat atau dibuatkan akun, hak akses pengguna tersebut akan ditentukan melalui menu Hak akses sebelum akun pengguna tersebut diaktivasi. Tampilan antar muka menu Hak akses disajikan pada Gambar 126, dan Gambar 127. Kode program untuk Gambar 126 – 127 bisa dilihat pada lampiran halaman 202 dengan judul “index.php”.

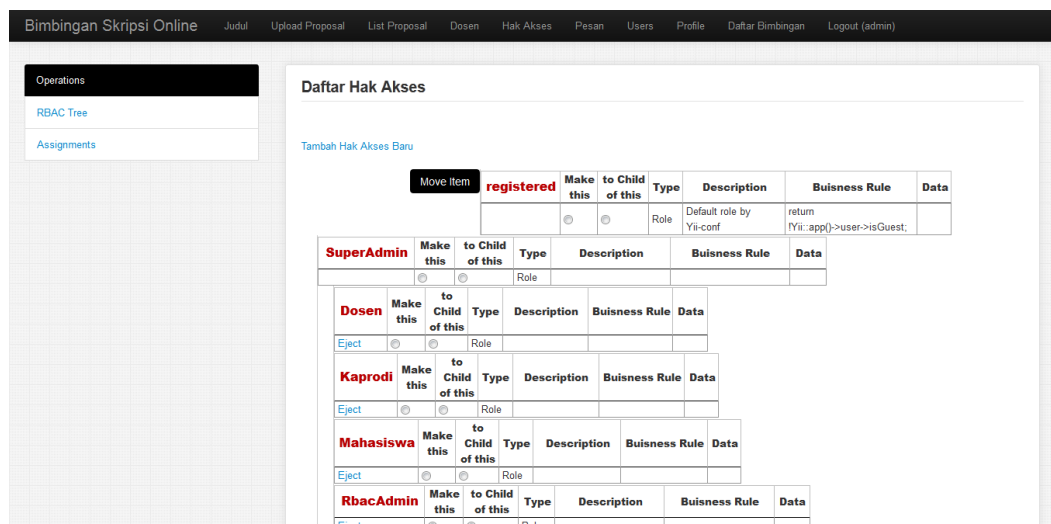


Gambar 126. Tampilan Antar Muka Menu Hak Akses (Administrator)

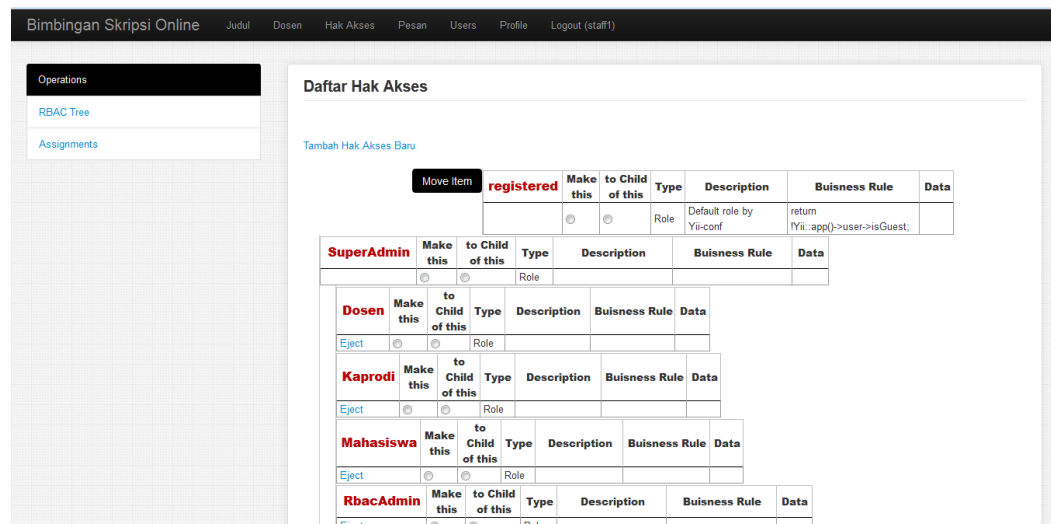


Gambar 127. Tampilan Antar Muka Menu Hak Akses (Staf)

Tampilan antar muka daftar hak akses disajikan pada Gambar 128, dan Gambar 129 menampilkan hak akses yang digunakan dalam sistem. Kode program untuk Gambar 128 – 129 bisa dilihat pada lampiran halaman 206 dengan judul “index.php”.



Gambar 128. Tampilan Antar Muka Menu Daftar Hak Akses (Administrator)



Gambar 129. Tampilan Antar Muka Menu Daftar Hak Akses (Staf)

Tampilan menu tambah hak akses disajikan pada Gambar 130, digunakan untuk membuat hak akses baru beserta deskripsi hak akses tersebut. Menu ini hanya bisa diakses oleh pengguna dengan hak akses Administrator. Kode program untuk

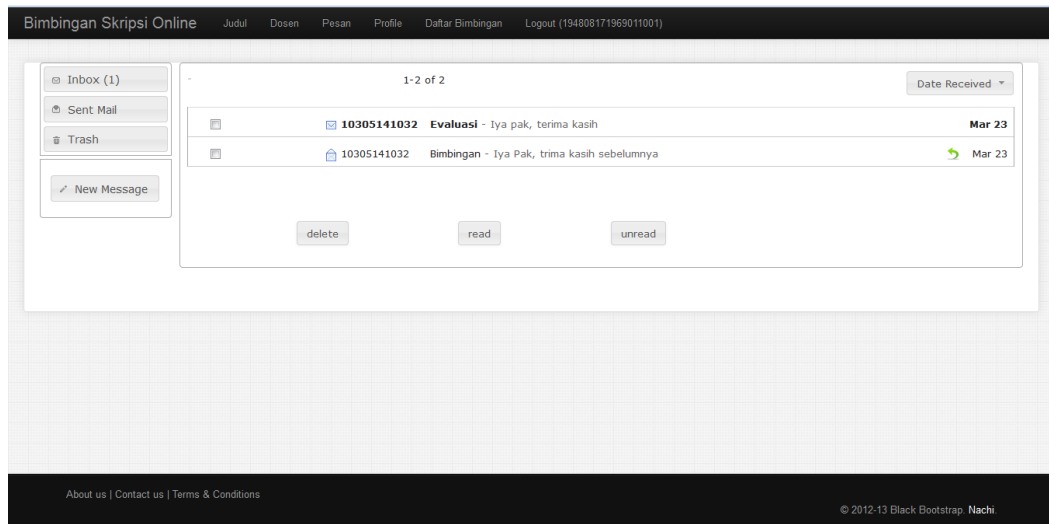
Gambar 130 bisa dilihat pada lampiran halaman 208 dengan judul “_editForm.php”.

Gambar 130. Tampilan Antar Muka Form Tambah Hak Akses

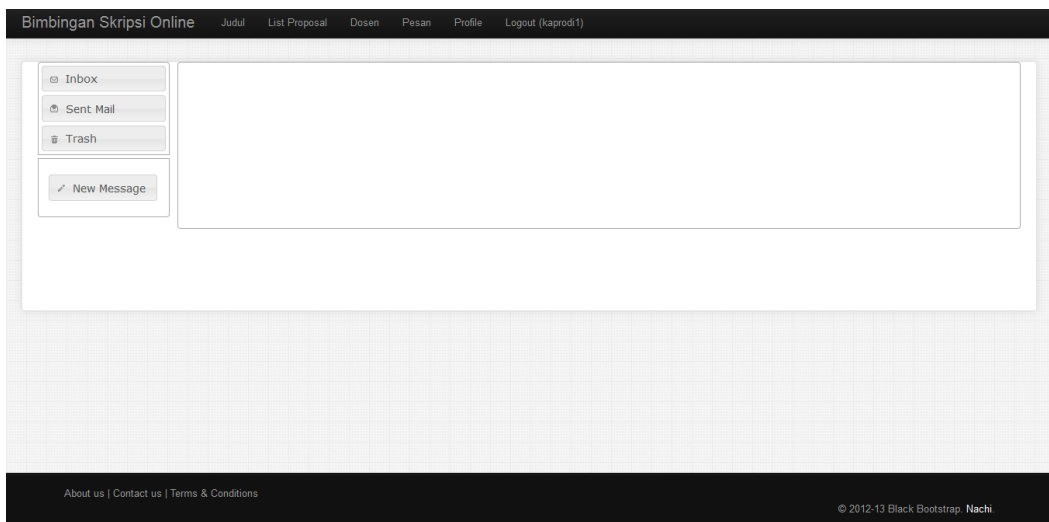
I. Tampilan Antar Muka Menu Pesan

Menu Pesan digunakan oleh pengguna untuk mengirim pesan ke pengguna lain. Menu ini digunakan untuk melakukan kegiatan bimbingan skripsi. Menu Pesan terdiri dari *New Message*, *Inbox*, *Sent Item*, dan *Trash*. Tampilan antar muka menu Pesan disajikan pada Gambar 131, Gambar 132, Gambar 133, Gambar 134, dan Gambar 135. Kode program untuk Gambar 131 – 135 bisa dilihat pada lampiran halaman 242 dengan judul “mailbox.php”.

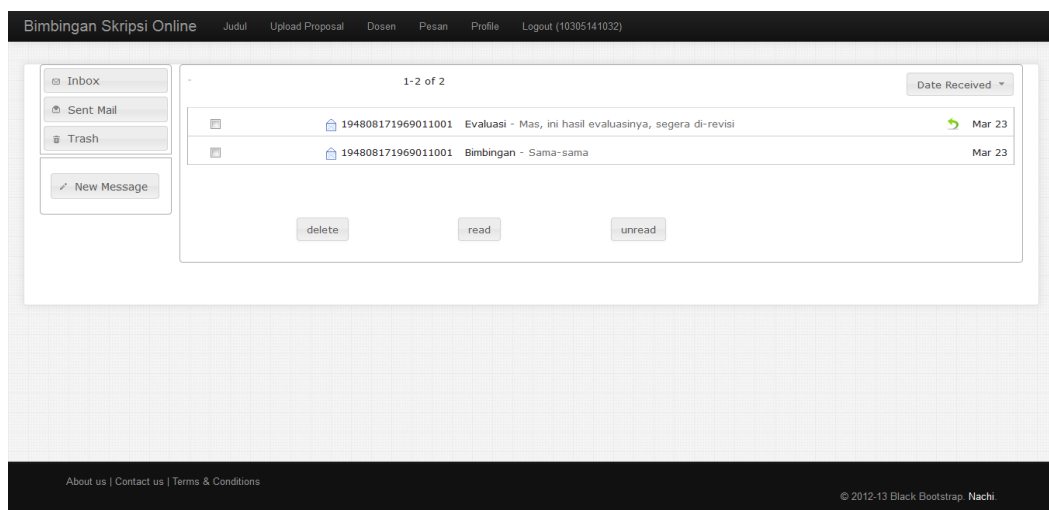
Gambar 131. Tampilan Antar Muka Menu Pesan (Administrator)



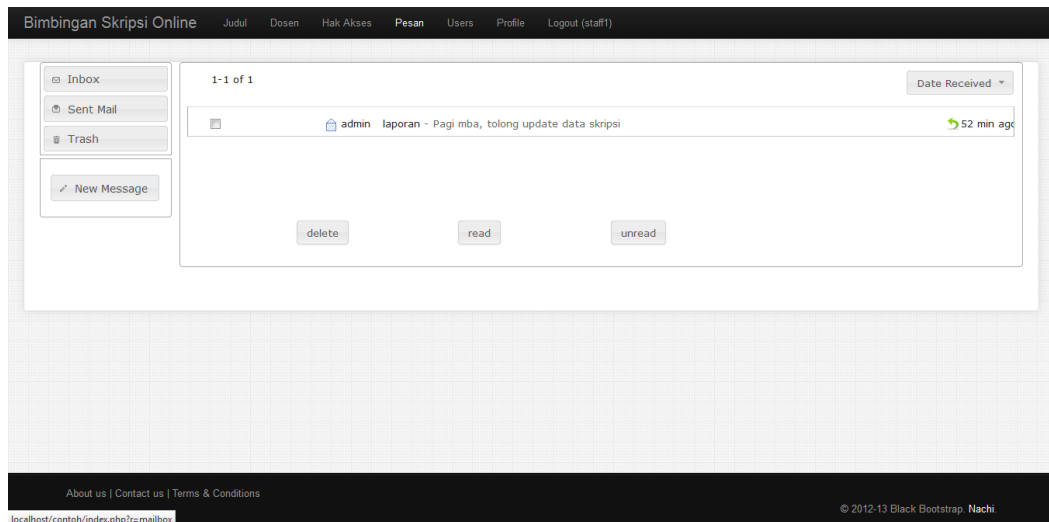
Gambar 132. Tampilan Antar Muka Menu Pesan (Dosen)



Gambar 133. Tampilan Antar Muka Menu Pesan (Kaprodi)

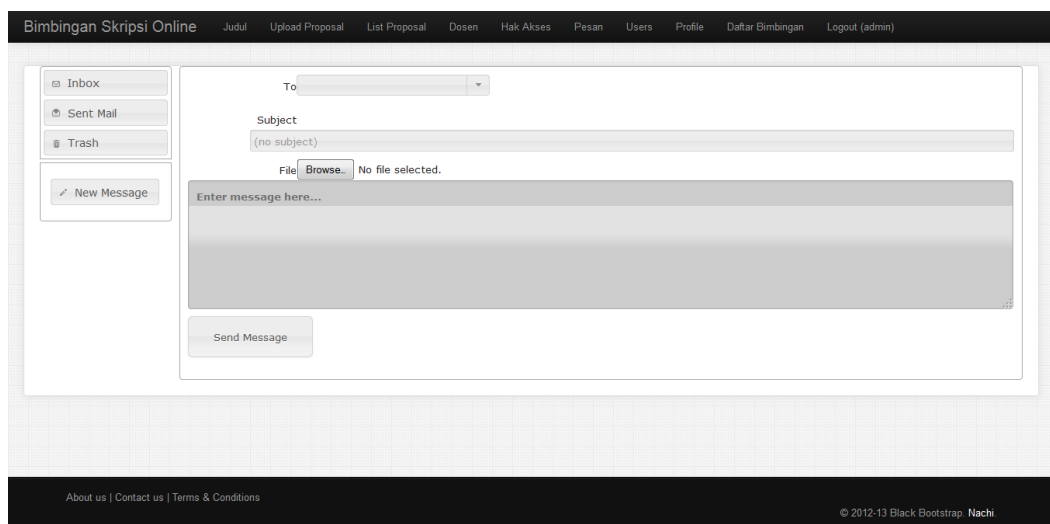


Gambar 134. Tampilan Antar Muka Menu Pesan (Mahasiswa)



Gambar 135. Tampilan Antar Muka Menu Pesan (Staf)

Tampilan *New Message* disajikan pada Gambar 136, Gambar 137, Gambar 138, Gambar 139, dan Gambar 140, terdiri dari 4 *field*, yaitu *to* (tujuan pengiriman), *Subject*, *File (attachment)*, dan Isi Pesan. Kode program untuk Gambar 136 – 140 bisa dilihat pada lampiran halaman 249 dengan judul “compose.php”.



Gambar 136. Tampilan Antar Muka Buat Pesan (Administrator)

The screenshot shows the 'Bimbingan Skripsi Online' web application. The top navigation bar includes links for 'Judul', 'Dosen', 'Pesan', 'Profile', 'Daftar Bimbingan', and 'Logout (194808171969011001)'. On the left sidebar, there are buttons for 'Inbox (1)', 'Sent Mail', 'Trash', and 'New Message'. The main content area is a form for composing a message. It includes a 'To' dropdown menu, a 'Subject' text field with the placeholder '(no subject)', a 'File' upload section with a 'Browse...' button and the text 'No file selected.', a large text area for the message body with the placeholder 'Enter message here...', and a 'Send Message' button at the bottom. The footer contains links for 'About us', 'Contact us', and 'Terms & Conditions', along with the copyright notice '© 2012-13 Black Bootstrap. Nachi.'.

Gambar 137. Tampilan Antar Muka Buat Pesan (Dosen)

This screenshot is identical to the previous one, showing the 'Bimbingan Skripsi Online' web application. The top navigation bar includes links for 'Judul', 'List Proposal', 'Dosen', 'Pesan', 'Profile', and 'Logout (kaprodi1)'. The left sidebar and the main message composition form are the same as in Gambar 137. The footer remains the same with 'About us', 'Contact us', 'Terms & Conditions', and the copyright notice '© 2012-13 Black Bootstrap. Nachi.'.

Gambar 138. Tampilan Antar Muka Buat Pesan (Kaprodi)

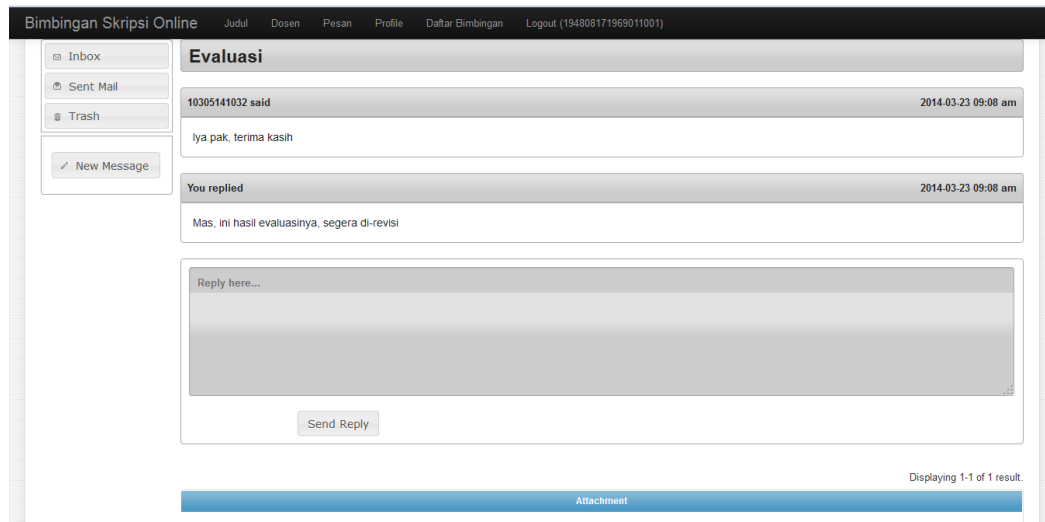
This screenshot is also identical to the previous ones, showing the 'Bimbingan Skripsi Online' web application. The top navigation bar includes links for 'Judul', 'Upload Proposal', 'Dosen', 'Pesan', 'Profile', and 'Logout (10305141032)'. The left sidebar and the main message composition form are consistent with the other two screenshots. The footer includes 'About us', 'Contact us', 'Terms & Conditions', and the copyright notice '© 2012-13 Black Bootstrap. Nachi.'.

Gambar 139. Tampilan Antar Muka Buat Pesan (Mahasiswa)

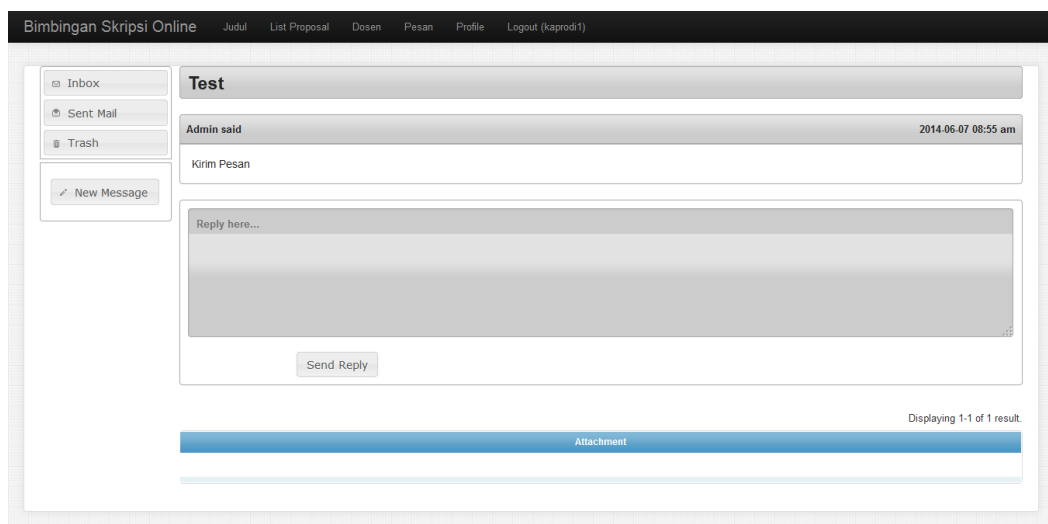
Gambar 140. Tampilan Antar Muka Buat Pesan (Staf)

Tampilan antar muka baca pesan disajikan pada Gambar 141, Gambar 142, Gambar 143, Gambar 144, dan Gambar 145, menampilkan percakapan dalam satu subjek pesan, *file attachment*, dan *field* balas pesan. Kode program untuk Gambar 141 – 145 bisa dilihat pada lampiran halaman 251 dengan judul “message.php”.

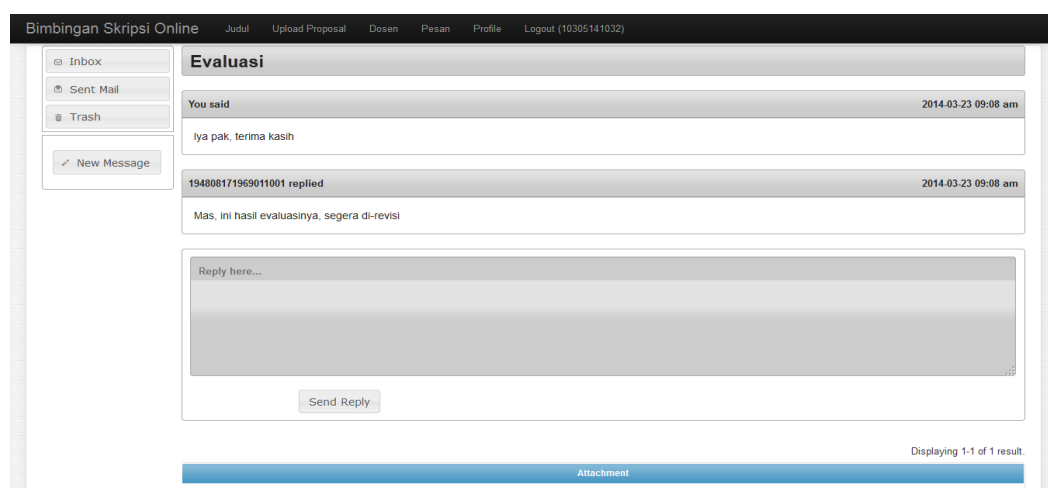
Gambar 141. Tampilan Antar Muka Baca Pesan (Administrator)



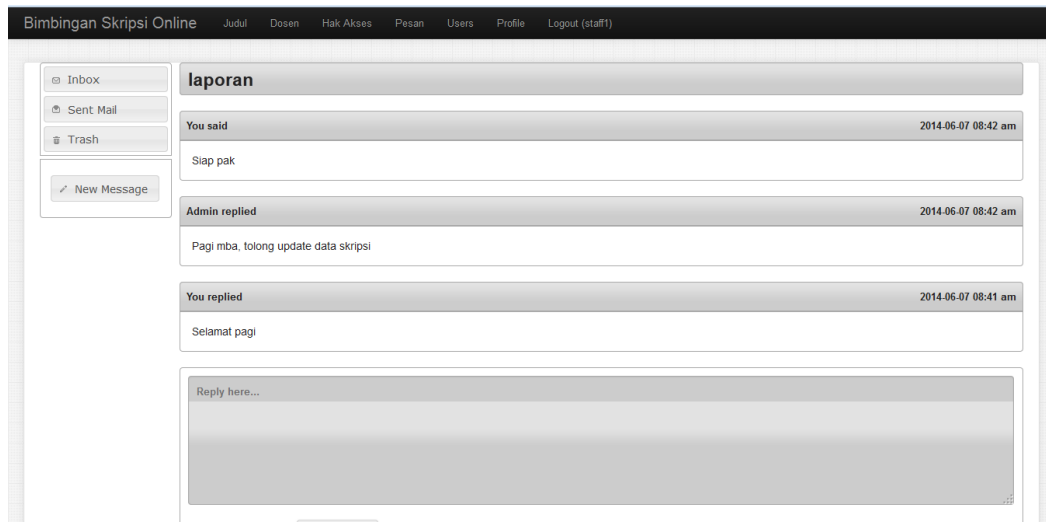
Gambar 142. Tampilan Antar Muka Baca Pesan (Dosen)



Gambar 143. Tampilan Antar Muka Baca Pesan (Kaprodi)



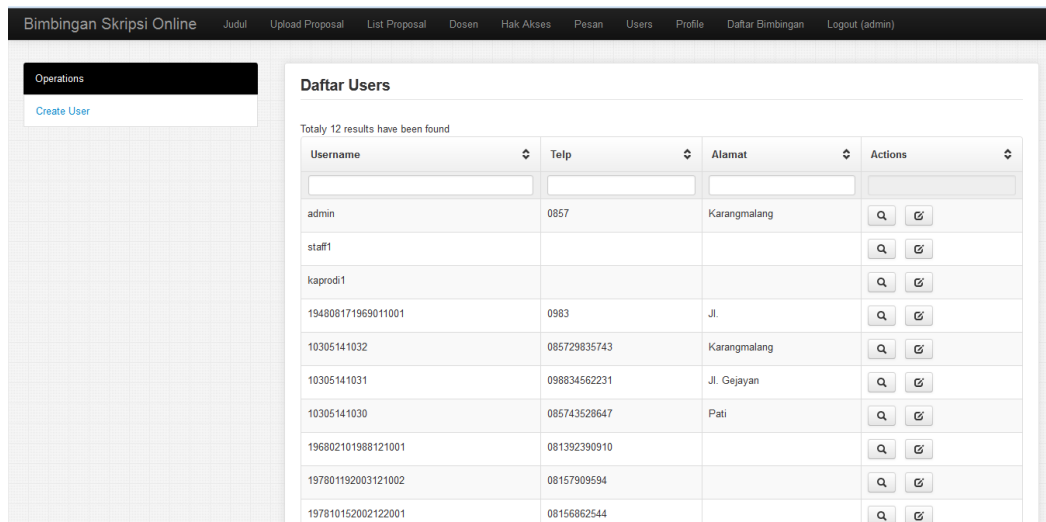
Gambar 144. Tampilan Antar Muka Baca Pesan (Mahasiswa)



Gambar 145. Tampilan Antar Muka Baca Pesan (Staf)

m. Tampilan Antar Muka Menu *Users*

Menu *Users* digunakan oleh Administrator atau Staf untuk mengelola data pengguna. Data yang ditampilkan adalah *Username* (NIM/NIP), *Email*, waktu akun dibuat, waktu terakhir pengguna *login*, dan Detail. Tampilan antar muka menu *Users* disajikan pada Gambar 146, dan Gambar 147. Kode program untuk Gambar 146 – 147 bisa dilihat pada lampiran halaman 264 dengan judul “index.php”.



Gambar 146. Tampilan Antar Muka Menu *Users* (Administrator)

Bimbingan Skripsi Online Judul Dosen Hak Akses Pesan Users Profile Logout (staff1)

Operations

[Create User](#)

Daftar Users

Totally 12 results have been found

Username	Telp	Alamat	Actions
admin	0857	Karangmalang	Q E
staff1			Q E
kaprodi1			Q E
194808171969011001	0983	Jl.	Q E
10305141032	085729835743	Karangmalang	Q E
10305141031	098834562231	Jl. Gejayan	Q E
10305141030	085743528647	Pati	Q E
196802101988121001	081392390910		Q E
197801192003121002	08157909594		Q E
197810152002122001	08156862544		Q E

Gambar 147. Tampilan Antar Muka Menu *Users* (Staf)

Tampilan detail pengguna (*View User*) disajikan pada Gambar 148, dan

Gambar 149, menampilkan data lengkap pengguna.

Bimbingan Skripsi Online Judul Upload Proposal List Proposal Dosen Hak Akses Pesan Users Profile Daftar Bimbingan Logout (admin)

Operations

[Create User](#)

[Update User](#)

[Delete User](#)

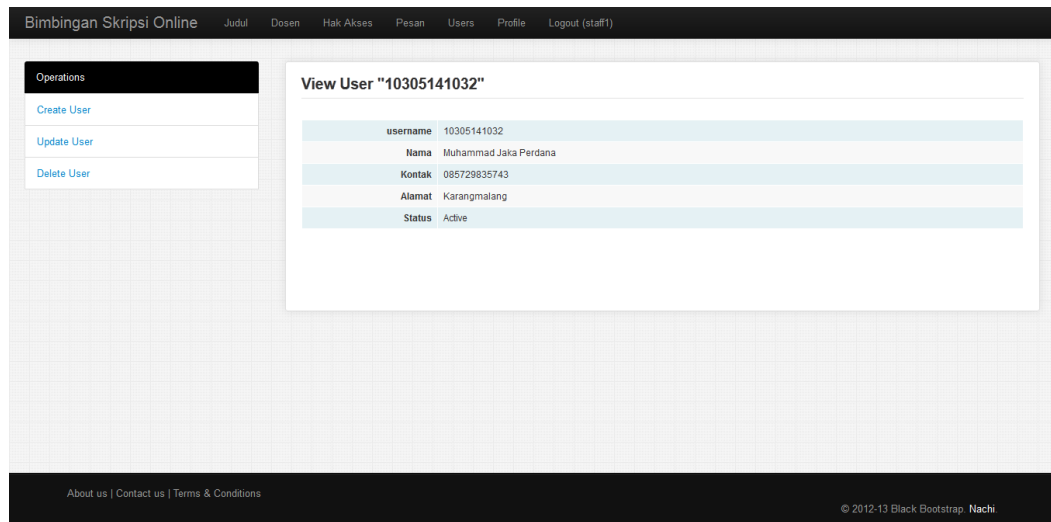
View User "10305141032"

username	10305141032
Nama	Muhammad Jaka Perdana
Kontak	085729835743
Alamat	Karangmalang
Status	Active

About us | Contact us | Terms & Conditions

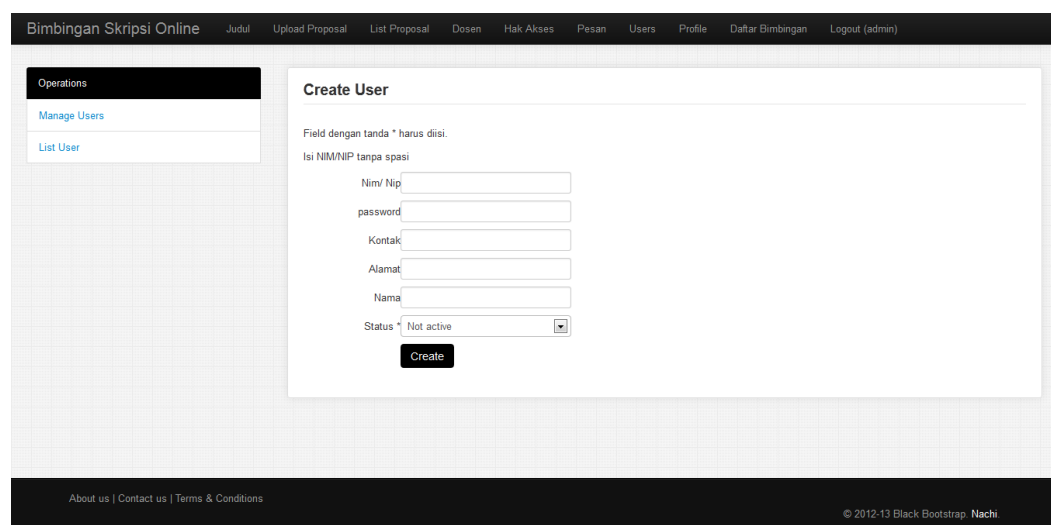
© 2012-13 Black Bootstrap. **Nachi**

Gambar 148. Tampilan Antar Muka Detail *Users* (Administrator)



Gambar 149. Tampilan Antar Muka Detail *Users* (Staf)

Menu tambah (*Create*) user disajikan pada Gambar 150, dan Gambar 151, digunakan oleh pengguna dengan hak akses Administrator dan Staf untuk menambah pengguna tanpa melalui pendaftaran. Menu ini terdiri dari 6 *field*, yaitu NIM/NIP, *password*, Nama, Alamat, Kontak, dan *Status*.



Gambar 150. Tampilan Antar Muka Form Tambah *Users* (Administrator)

The screenshot displays the 'Bimbingan Skripsi Online' web interface. At the top, a dark navigation bar contains the title and links for Judul, Dosen, Hak Akses, Pesan, Users, Profile, and Logout (staff). On the left, a sidebar under the 'Operations' header lists 'Manage Users' and 'List User'. The main content area is titled 'Create User' and includes a note: 'Field dengan tanda * harus diisi.' Below this, a form is provided for creating a new user, with the instruction 'Isi NIM/NIP tanpa spasi'. The form fields are: 'Nim/ Nip', 'password', 'Kontak', 'Alamat', 'Nama', and 'Status *' (a dropdown menu currently showing 'Not active'). A 'Create' button is positioned at the bottom of the form. The footer of the page contains links for 'About us | Contact us | Terms & Conditions' and a copyright notice: '© 2012-13 Black Bootstrap. Nachi.'

Gambar 151. Tampilan Antar Muka Form Tambah *Users* (Staf)

Berdasarkan fitur-fitur aplikasi yang telah dirancang, Aplikasi Bimbingan Skripsi *Online* dapat membantu proses pembuatan skripsi di Jurdikmat UNY di beberapa bagian, yaitu:

1. Dokumentasi data skripsi.
2. Pencarian judul skripsi yang telah digunakan di Jurdikmat UNY.
3. Pengajuan proposal skripsi.
4. Pengecekan daftar bimbingan.
5. Pengiriman pesan/*file* untuk proses bimbingan.

D. Pengujian Sistem

Pengujian sistem yang dilakukan adalah pengujian *Betha* yang dilakukan secara objektif. Pengujian dilakukan dengan kuesioner yang disusun berdasarkan faktor-faktor jaminan kualitas perangkat lunak.

Pengujian dilakukan kepada 10 pengguna yang terbagi ke dalam 4 tipe pengguna (5 mahasiswa, 2 dosen, 1 kaprodi, dan 2 staf) dan seorang ahli komputer.

Dari hasil pengujian kuesioner tersebut, dilakukan perhitungan untuk dapat menentukan kesimpulan terhadap penilaian aplikasi yang dibangun.

Kuesioner untuk pengguna terdiri atas 24 pertanyaan yang disusun berdasarkan faktor-faktor yang berkaitan dengan sifat-sifat operasional software, yaitu faktor-faktor yang menunjukkan atribut kualitas produk dilihat dari sudut pandang pengguna (*Correctness*, *Reliability*, *Integrity*, dan *Usability*), dengan 4 pilihan jawaban, yaitu Sangat Tidak Setuju (STS), Tidak Setuju (TS), Setuju (S), dan Sangat Setuju (SS). Berikut adalah daftar pertanyaan pada kuesioner:

A. *Correctness*

1. Menu yang tersedia memenuhi kebutuhan.
2. Fitur-fitur yang tersedia di Aplikasi Bimbingan Skripsi *Online* membantu proses pembuatan skripsi.
3. Butir-butir informasi pada menu Judul memenuhi kebutuhan.
4. Menu Upload Proposal memudahkan mahasiswa untuk mengajukan proposal skripsi kepada Ketua Program Studi tanpa harus bertatap muka.
5. Menu Pesan efektif untuk membantu proses bimbingan skripsi.
6. Menu Dosen (data dosen) diperlukan bagi dosen.
7. Menu Dosen (data dosen) diperlukan bagi Ketua Program Studi.
8. Fitur download proposal membantu Ketua Program Studi untuk menerima file proposal skripsi dari mahasiswa tanpa harus bertatap muka.

B. *Reliability*

9. Pencarian judul skripsi pada menu Judul (judul skripsi) menampilkan hasil yang sesuai dengan kata kunci yang dimasukkan.

10. Pencarian data dosen pada menu Dosen (data dosen) akurat.
11. Pesan yang dibuat pada menu Pesan sesuai dengan pesan yang terkirim.
12. Data diri yang ditampilkan sesuai dengan data yang dimasukkan saat mendaftar.
13. Menu Judul (judul skripsi) menampilkan informasi yang sesuai dengan data yang dimasukkan.
14. Hak akses pengguna yang ditampilkan sesuai dengan data hak akses pengguna yang dimasukkan.

C. *Integrity*

15. Data yang dimasukkan aman dari orang yang tidak berhak mengakses.

D. *Usability*

16. Struktur menu dan fungsi dalam sistem mudah dipahami.
17. Instruksi atau petunjuk yang tersedia membantu menggunakan aplikasi.
18. Bahasa yang digunakan mudah dipahami.
19. Tulisan di semua halaman dapat terbaca dengan jelas.
20. Tampilan Aplikasi Bimbingan Skripsi *Online* menarik.
21. Pengelolaan data (tambah, edit, hapus data) pada menu Judul mudah dilakukan.
22. Pengelolaan data (tambah, edit, hapus data) pada menu Dosen mudah dilakukan.
23. Menu Hak akses mudah digunakan.
24. Pengelolaan data (tambah, edit, hapus data) pada menu Users (data pengguna) mudah dilakukan.

Berdasarkan hasil pengujian *betha*, dicari persentase masing-masing jawaban dengan menggunakan rumus:

$$Y = (P/Q) \times 100\%$$

Keterangan :

Y = Nilai persentase

P = Banyaknya jawaban responden tiap soal

Q = Jumlah responden

Tabel 11. Persentase Hasil Pengujian *Betha* Kepada Pengguna

Pertanyaan	Total Responden	STS	TS	S	SS
<i>Correctness</i>					
Menu yang tersedia memenuhi kebutuhan	10		20%	60%	20%
Fitur-fitur yang tersedia di Aplikasi Bimbingan Skripsi <i>Online</i> membantu proses pembuatan skripsi	10			50%	50%
Butir-butir informasi pada menu Judul memenuhi kebutuhan	5			100%	
Menu Upload Proposal memudahkan mahasiswa untuk mengajukan proposal skripsi kepada Ketua Program Studi tanpa harus bertatap muka	5			40%	60%
Menu Pesan efektif untuk membantu proses bimbingan skripsi	9			67%	33%
Menu Dosen (data dosen) diperlukan bagi dosen	2		100%		
Menu Dosen (data dosen) diperlukan bagi Ketua Program Studi	1				100%
Fitur download proposal membantu Ketua Program Studi untuk menerima file proposal skripsi dari mahasiswa tanpa harus bertatap muka	1				100%
<i>Reliability</i>					
Pencarian judul skripsi pada menu Judul (judul skripsi) menampilkan hasil yang sesuai	8			25%	75%

dengan kata kunci yang dimasukkan					
Pencarian data dosen pada menu Dosen (data dosen) akurat	5			40%	60%
Pesan yang dibuat pada menu Pesan sesuai dengan pesan yang terkirim	9			33%	67%
Data diri yang ditampilkan sesuai dengan data yang dimasukkan saat mendaftar	9			44%	56%
Menu Judul (judul skripsi) menampilkan informasi yang sesuai dengan data yang dimasukkan	3			33%	67%
Hak akses pengguna yang ditampilkan sesuai dengan data hak akses pengguna yang dimasukkan	2			50%	50%
<i>Integrity</i>					
Data yang dimasukkan aman dari orang yang tidak berhak mengakses	10			60%	40%
<i>Usability</i>					
Struktur menu dan fungsi dalam sistem mudah dipahami	10			40%	60%
Instruksi/petunjuk yang tersedia membantu menggunakan aplikasi	10			50%	50%
Bahasa yang digunakan mudah dipahami	8			25%	75%
Tulisan di semua halaman dapat terbaca dengan jelas	8			25%	75%
Tampilan Aplikasi Bimbingan Skripsi <i>Online</i> menarik	10		20%	80%	
Pengelolaan data (tambah, edit, hapus data) pada menu Judul mudah dilakukan	3			67%	33%
Pengelolaan data (tambah, edit, hapus data) pada menu Dosen mudah dilakukan	2			50%	50%
Menu Hak akses mudah digunakan	2			50%	50%
Pengelolaan data (tambah, edit, hapus data) pada menu Users (data pengguna) mudah dilakukan	2			50%	50%

Sedangkan kuesioner untuk ahli komputer terdiri atas 12 pertanyaan yang disusun berdasarkan faktor-faktor yang tidak diujikan kepada pengguna (*Efficiency, Maintainability, Testability, Flexibility, Portability, Reusability, Interopability*). Hasil pengujian kepada ahli komputer dapat dilihat pada Tabel 12.

Tabel 12. Persentase Hasil Pengujian Kepada Ahli Komputer

Keterangan	STS	TS	S	SS
<i>Efficiency</i>				
Penggunaan <i>code</i> pada sistem efisien			√	
<i>Maintainability</i>				
Perbaikan data skripsi mudah dilakukan			√	
Perbaikan data pengguna mudah dilakukan			√	
<i>Testability</i>				
Pencarian judul skripsi pada menu Judul (judul skripsi) menampilkan hasil yang sesuai dengan kata kunci yang dimasukkan				√
Menu Judul (judul skripsi) menampilkan informasi yang sesuai dengan data yang dimasukkan				√
Pesan yang dibuat pada menu pesan sesuai dengan pesan yang terkirim				√
File proposal yang di- <i>upload</i> pada menu Upload Proposal muncul pada list download proposal pada menu List Proposal				√
<i>Flexibility</i>				
Sistem yang dibuat dengan bahasa pemrograman PHP dapat dimodifikasi dengan bahasa pemrograman lain (<i>javascript</i>)			√	
<i>Portability</i>				
Sistem dapat ditampilkan/diakses di beberapa hardware yaitu komputer, tablet, dan <i>smartphone</i>				√
Sistem dapat ditampilkan/diakses di beberapa <i>browser</i> yaitu mozilla firefox, dan google chrome				√
<i>Reusability</i>				
Data yang tersimpan di database dapat diakses melalui di Ms. excel			√	
<i>Interoperability</i>				
Proses pendaftaran pengguna (Mahasiswa) berhubungan dengan proses <i>upload</i> proposal			√	

Tabel 13. Keterangan Kelayakan, Saran, Dan Revisi Aplikasi

No	Responden	Keterangan Layak	Saran	Keterangan Revisi	Hambatan Revisi
1	Mahasiswa	Layak	Hasil pencarian pada filter ditambahkan highlight pada kata yang dimasukkan	Belum	Keterbatasan penulis
2	Mahasiswa	Layak	Data profil ditambahkan kontak	Sudah	-
3	Mahasiswa	Layak	Data profil ditambahkan kontak	Sudah	-
4	Mahasiswa	Layak	-	-	-
5	Mahasiswa	Layak	1. Kurangi ruang kosong 2. Font diperbesar	1. Belum 2. Sudah	1. Konten sedikit agar pengguna tidak bingung
6	Dosen	Layak	Tambahkan daftar bimbingan bagi dosen	Sudah	-
7	Dosen	Layak	1. Tambahkan daftar bimbingan bagi dosen 2. Sistem dihubungkan ke siakad	1. Sudah 2. Belum	1. - 2. Keterbatasan penulis
8	Staf	Layak	Tampilan dibuat <i>full color</i>	Belum	
9	Staf	Layak	Tambahkan daftar penguji menjadi 4 pada data skripsi	Sudah	-
10	Kaprodi	Layak	Dibuat berdasarkan prosedur skripsi Jurdikmat	Sudah	-

11	Ahli Komputer	Layak	1. Tambahkan daftar bimbingan bagi dosen 2. Buat kontak pesan bagi pengguna	1. Sudah 2. Belum	1. - 2. Keterbatasan penulis
----	---------------	-------	--	--------------------------	-------------------------------------

Berdasarkan hasil pengujian *betha* yang telah dilakukan dapat ditarik kesimpulan bahwa program Aplikasi Bimbingan Skripsi *Online* mudah dipelajari, cukup mudah digunakan, informasi yang dihasilkan cukup akurat, dan cukup membantu dalam proses bimbingan skripsi, pengelolaan dan dokumentasi data skripsi.

BAB IV

PENUTUP

A. Kesimpulan

1. Aplikasi Bimbingan Skripsi *Online* dirancang untuk membantu proses bimbingan skripsi mahasiswa Jurdikmat UNY. Rancangan Aplikasi Bimbingan Skripsi *Online* adalah sebagai berikut:

a. Terdiri dari 5 tipe pengguna, yaitu:

- 1) Administrator, merupakan staf Jurdikmat yang mempunyai kewenangan penuh terhadap fungsi sistem.
- 2) Dosen, yaitu semua dosen aktif pembimbing skripsi di Jurdikmat UNY.
- 3) Kaprodi, yaitu ketua program studi matematika dan ketua program studi pendidikan matematika Jurdikmat UNY yang sedang bertugas.
- 4) Mahasiswa, yaitu mahasiswa aktif Jurdikmat UNY yang telah menyelesaikan seluruh syarat pengambilan TA.
- 5) Staf, yaitu staf aktif Jurdikmat UNY yang mengelola kebutuhan administrasi tugas akhir skripsi mahasiswa Jurdikmat UNY.

b. Terdiri dari 10 menu, yaitu:

- 1) *Login/Logout*, berfungsi untuk masuk/keluar aplikasi.
- 2) Judul, berfungsi untuk mendokumentasikan dan melakukan pencarian data judul skripsi.

- 3) Upload Proposal, berfungsi untuk mengajukan proposal skripsi dari mahasiswa kepada Kaprodi.
 - 4) List Proposal, berfungsi untuk melihat daftar proposal yang diajukan oleh mahasiswa.
 - 5) Dosen, berfungsi untuk melihat data dosen Jurdikmat UNY.
 - 6) Hak akses, berfungsi untuk menentukan hak akses pengguna yang mendaftar.
 - 7) Pesan, berfungsi untuk melakukan bimbingan dan pemberitahuan.
 - 8) *Users*, berfungsi untuk mengelola data pengguna.
 - 9) Daftar Bimbingan, berfungsi untuk melihat daftar mahasiswa bimbingan.
 - 10) Profil, berfungsi untuk melihat data pribadi.
2. Berdasarkan fitur-fitur aplikasi yang telah dirancang, Aplikasi Bimbingan Skripsi *Online* dapat membantu proses pembuatan skripsi di Jurdikmat UNY di beberapa bagian, yaitu dokumentasi data skripsi, pencarian judul skripsi yang telah digunakan di Jurdikmat UNY, pengajuan proposal skripsi, pengecekan daftar bimbingan, pengiriman pesan/*file* untuk proses bimbingan.

B. Saran

Berikut adalah beberapa saran untuk menyempurnakan Aplikasi Bimbingan Skripsi *Online* agar aplikasi bisa lebih baik:

1. Aplikasi diujikan secara nyata dengan mengadakan pemakaian program dalam lingkup terbatas sehingga pengguna merasakan kelebihan dan kekurangan program secara langsung.
2. Tampilan dibuat lebih menarik.
3. Pada hasil pencarian judul skripsi, akan lebih baik jika kata yang sesuai dengan kata kunci yang dimasukkan ditandai sehingga pengguna bisa mengetahui bagian mana dari hasil pencarian yang sesuai dengan kata kunci.
4. Sistem dihubungkan dengan siakad.

DAFTAR PUSTAKA

- Abdul Kadir. (1999). *Konsep dan Tuntunan Praktis Basis Data*. Yogyakarta: Penerbit ANDI.
- Abdul Kadir. (2001). *Dasar Pemrograman WEB Dinamis Menggunakan PHP*. Yogyakarta: Penerbit ANDI.
- Abdul Kadir. (2003). *Pengenalan Sistem Informasi*. Yogyakarta: Penerbit Andi.
- Agus Winarno. (2007). *Analisa & Perancangan Sistem Informasi*. Yogyakarta. Universitas Gajah Mada.
- Al Fatta, H. (2007). *Analisis dan Perancangan Sistem Informasi untuk Keunggulan Organisasi Perusahaan dan Organisasi Modern*. Ed 1. Yogyakarta: Penerbit ANDI.
- Al-Bahra bin Ladjamuddin. (2005). *Analisis dan Desain Sistem Informasi*. Yogyakarta: Penerbit Graha Media.
- Andri Kristanto. (2003). *Perancangan Sistem Informasi dan Aplikasinya*. Yogyakarta: Penerbit Gava media.
- Anik Ghufon, dkk. (2013). *Pedoman Penulisan Tugas Akhir*. Yogyakarta: Universitas Negeri Yogyakarta.
- Aprillita Dwiyan. (2013). *Perancangan Sistem Pendukung Bimbingan Online Tugas Akhir Mahasiswa Program Studi Teknik Informatika*. Diakses dari <http://jurnal.mti.cs.ui.ac.id/index.php/jsi/article/download/346/218> pada tanggal 15 Maret 2014, Jam 11.43 WIB.
- Badiyanto. (2013). *Buku Pintar Framework Yii*. Yogyakarta: Mediakom.
- Deni Sutaji. (2012). *Sistem Inventory Mini Market Dengan PHP dan jQuery*. Yogyakarta: Penerbit Lokomedia.
- Firman Kemal P. (2010). *Sistem Informasi Simpan Pinjam Di Koperasi Karyawan Pusat Perencanaan Dan Pengembangan Perumka "Dinamiko" Di PT. Kereta Api Indonesia (Persero) Bandung*. Diakses dari <http://elib.unikom.ac.id/gdl.php?mod=browse&op=read&id=jbptunikompp-gdl-firmankema-22183> pada tanggal 15 Maret 2014, Jam 16.20 WIB.
- Laudon, Kenneth C. & Laudon, Jane P. (2008) *Sistem Informasi Manajemen*. Palgrave: Basingstoke.
- Nastiti. (2012). *Sistem Informasi Transaksi Di LIMUNY Lounge*. Skripsi. FT UNY.
- Pressman, Roger S. (2001). *Software Engineering: A Practitioner's Approach*. New York: McGraw-Hill.
- Syahril Akbar. (2010). *Sistem Informasi Pengolahan Data Honor Wartawan Dan Penulis PT. Galamedia Bandung Perkasa*. Diakses dari

<http://elib.unikom.ac.id/gdl.php?mod=browse&op=read&id=jbptunikompp-gdl-syahrilakb-22180> pada tanggal 15 Maret 2014, Jam 16.05 WIB.

Wisnu. (2012). *Software Requirements Specification*. Diakses dari <http://cisini.wordpress.com/2012/10/16/srs/> pada tanggal 21 Maret 2014, Jam 15.30 WIB.

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

1. Pendahuluan

1.1 Tujuan

Dokumen ini berisi Spesifikasi Kebutuhan Perangkat Lunak (SKPL) atau *Software Requirement Specification* (SRS) untuk Aplikasi Bimbingan Skripsi *Online*. Tujuan dari penulisan dokumen ini adalah untuk memberikan penjelasan mengenai perangkat lunak yang akan dibangun baik berupa gambaran umum maupun penjelasan detail dan menyeluruh.

1.2 Lingkup Masalah

Dokumen ini memberikan deskripsi dan gambaran dari kebutuhan *user* terhadap Aplikasi Bimbingan Skripsi *Online*.

1.3 Definisi, Akronim, Singkatan yang Dipakai

Istilah	Keterangan
1. User	User adalah orang yang menggunakan aplikasi
Sistem	Kumpulan komponen-komponen yang saling berkesinambungan untuk mencapai tujuan.

1.4 Referensi

1. IEEE Std 830-1993, IEEE *Recommended Practice for Software Requirement Specifications*.
2. SOP Pembuatan tugas akhir skripsi Jurusan Pendidikan Matematika Universitas Negeri Yogyakarta.

1.5 Gambaran Umum Dokumen

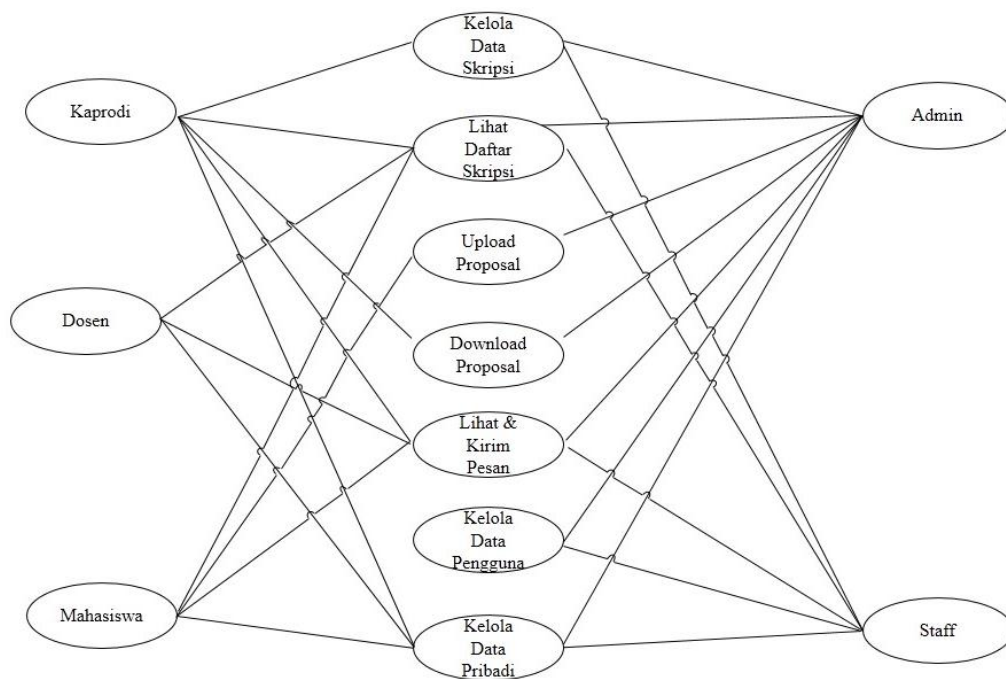
Dokumen ini secara garis besar terdiri dari tiga bab dengan perincian sebagai berikut:

1. Bab 1 Pendahuluan, merupakan pengantar dokumen SRS yang berisi tujuan penulisan dokumen, lingkup masalah pengembangan perangkat lunak, juga memuat definisi, akronim dan istilah yang digunakan serta deskripsi umum dokumen yang merupakan ikhtisar dokumen SRS.
2. Bab 2 Deskripsi Global Perangkat Lunak, mendefinisikan perspektif produk perangkat lunak serta asumsi dan ketergantungan yang digunakan dalam pengembangan sistem bimbingan skripsi *online*.
3. Bab 3 Deskripsi Rinci Kebutuhan, mendeskripsikan kebutuhan khusus bagi sistem bimbingan skripsi *online*, yang meliputi kebutuhan antarmuka, kebutuhan fungsionalitas, kebutuhan performansi, batasan perancangan, atribut sistem perangkat lunak dan kebutuhan lain dari sistem.

2. Deskripsi Global Perangkat Lunak

2.1 Deskripsi Umum

Aplikasi ini bertujuan mendukung proses bimbingan skripsi agar bisa berjalan tanpa harus tatap muka dan mendokumentasikan informasi tugas akhir agar lebih tertata dan lebih mudah diakses. Aplikasi terdiri dari 5 tipe *user* yaitu Administrator, Dosen, Kaprodi, Mahasiswa, dan Staf. Setiap tipe *user* memiliki hak akses berbeda dalam aplikasi. *User* yang bisa mengakses menu dalam aplikasi hanya *user* yang telah *login*.



2.2 Fungsi Produk

Produk ini memiliki fungsi-fungsi sebagai berikut.

1. Lihat informasi skripsi

Deskripsi : melihat data skripsi yang telah dibuat mahasiswa

Aktor : semua *user*

Input : -

2. Ubah data skripsi

Deskripsi : mengubah data skripsi dalam sistem

Aktor : Staf, Kaprodi

Input : -

3. *Upload* proposal

Deskripsi : proses mengajukan proposal judul skripsi

Aktor : Mahasiswa

Input : judul skripsi, file proposal

4. *Download* proposal

Deskripsi : proses *download* file proposal

Aktor : Kaprodi

Input : -

5. Lihat data dosen

Deskripsi : melihat data dosen yang ada di Jurdikmat UNY

Aktor : semua *user*

Input : -

6. Ubah data dosen

Deskripsi : mengubah data dosen dalam sistem

Aktor : Staf

Input : -

7. Hak akses

Deskripsi : menentukan hak akses pengguna

Aktor : Staf

Input : -

8. Pesan

Deskripsi : fungsi yang digunakan untuk melakukan proses bimbingan skripsi, atau melakukan pemberitahuan kepada user

Aktor : semua *user*

Input : NIM/NIP, subjek, file dokumen, isi pesan

9. Data pengguna

Deskripsi : mengakses dan mengubah status pengguna

Aktor : Staf

Input : status pengguna

10. Profil

Deskripsi : melihat dan mengubah data pribadi

Aktor : semua *user*

Input : data diri

2.3 Karakteristik *User*

Type User	Hak	Kewajiban
Administrator	<ul style="list-style-type: none">- Mengakses semua fitur dalam sistem- Mengubah struktur atau tampilan aplikasi	<ul style="list-style-type: none">- Melakukan perbaikan jika ada kerusakan- Melakukan perawatan dan menjaga performa aplikasi
Dosen	<ul style="list-style-type: none">- Melakukan proses bimbingan skripsi melalui aplikasi	<ul style="list-style-type: none">- Menanggapi mahasiswa yang melakukan bimbingan skripsi melalui sistem
Kaprodi	<ul style="list-style-type: none">- Mengakses file proposal skripsi	<ul style="list-style-type: none">- Melakukan verifikasi proposal skripsi- Melakukan pemberitahuan kepada mahasiswa, dosen, atau staf
Mahasiswa	<ul style="list-style-type: none">- <i>Upload</i> proposal skripsi- Melakukan proses bimbingan skripsi melalui aplikasi	<ul style="list-style-type: none">- Mengecek judul skripsi yang ada dalam sistem sebelum meng-<i>upload file</i> skripsi
Staf	<ul style="list-style-type: none">- Mengubah data daftar skripsi, dosen, dan mahasiswa	<ul style="list-style-type: none">- Melakukan aktivasi akun pengguna- Menentukan hak akses pengguna- Menambah data skripsi atau dosen jika ada- Mengubah data jika ada kesalahan atau perubahan- Menonaktifkan akun pengguna yang sudah tidak boleh mengakses aplikasi

2.4 Batasan Masalah

1. Aplikasi dibuat dalam lingkup Jurusan Pendidikan Matematika UNY.
2. Tidak membahas aktivitas yang dilakukan secara *offline*.

2.5 Asumsi dan Ketergantungan

1. Semua aktivitas yang dilakukan secara *offline* dianggap telah diselesaikan.
2. *Tool* pengembangan telah tersedia, dalam hal ini yang dimaksud dengan *tool* adalah *software* dan *hardware* yang dibutuhkan untuk pengembangan.
3. Informasi tentang tugas akhir dalam sistem dibuat berdasarkan data dari Jurusan Pendidikan Matematika UNY.

2. Spesifikasi Kebutuhan

3.1 Kebutuhan Antarmuka Pengguna

Aplikasi ini menggunakan antarmuka berbasis *web* dan perangkat keras berupa komputer. Pengguna mengoperasikan aplikasi menggunakan *keyboard* dan *mouse* dengan sistem operasi windows.

3.2 Kebutuhan Antarmuka *Hardware*

Aplikasi bimbingan skripsi ini berjalan dengan perangkat keras komputer dengan *keyboard* dan *mouse* sebagai alat untuk mengoperasikan aplikasi. Masing-masing perangkat keras merupakan komponen penting dalam aplikasi ini.

3.3 Kebutuhan Antarmuka *Software*

Aplikasi dirancang dengan bahasa pemrograman *web*, PHP, *software* *Sublime* 2.0 sebagai *text editor*, MySQL Xampp 7 sebagai sistem manajemen

database, *Mozilla Firefox* sebagai *web browser* dan *framework Yii* sebagai kerangka kerja.

3.4 Kebutuhan Antarmuka Komunikasi

Aplikasi ini merupakan aplikasi yang menggunakan *internet* sebagai sarana komunikasi antara komponen.

3.5 Kebutuhan Fungsionalitas

Berikut adalah penjelasan lebih detail tentang fungsi-fungsi produk. Detail meliputi kondisi awal, alur kerja, kondisi akhir, pengecualian, dan kebutuhan non-fungsional.

1. Lihat informasi skripsi

Nama Fungsi	Lihat informasi skripsi
Kondisi awal	Aktor telah terotentikasi
Alur kerja	1. Klik menu Judul 2. Sistem menampilkan daftar skripsi 3. Klik detail pada skripsi 4. Sistem menampilkan detail skripsi
Kondisi akhir	Detail informasi skripsi
Pengecualian	-
Kebutuhan non-fungsional	Fungsi ini bisa dipakai semua <i>user</i> yang <i>login</i>

2. Ubah data skripsi

Nama Fungsi	Ubah data skripsi
Kondisi awal	Aktor telah terotentikasi sebagai Staf atau Kaprodi
Alur kerja	1. Klik menu Judul 2. Sistem menampilkan daftar skripsi 3. Klik edit 4. Sistem menampilkan form data skripsi 5. Ubah data 6. Simpan perubahan
Kondisi akhir	Data berubah
Pengecualian	-
Kebutuhan non-fungsional	Fungsi ini hanya bisa dipakai oleh Staf dan Kaprodi

3. *Upload* proposal

Nama Fungsi	Ubah data skripsi
Kondisi awal	Aktor telah terotentikasi sebagai Mahasiswa
Alur kerja	<ol style="list-style-type: none"> 1. Klik menu <i>Upload</i> Proposal 2. Sistem menampilkan form <i>Upload</i> Proposal 3. Masukkan judul dan file proposal 4. <i>Upload</i> data
Kondisi akhir	Data ter- <i>upload</i>
Pengecualian	-
Kebutuhan non-fungsional	Fungsi ini hanya bisa dipakai oleh Mahasiswa

4. *Download* proposal

Nama Fungsi	<i>Download</i> proposal
Kondisi awal	Aktor telah terotentikasi sebagai Kaprodi
Alur kerja	<ol style="list-style-type: none"> 1. Klik menu List Proposal 2. Sistem menampilkan daftar proposal yang telah di-<i>upload</i> 3. Klik nama file proposal
Kondisi akhir	Data ter- <i>download</i>
Pengecualian	-
Kebutuhan non-fungsional	Fungsi ini hanya bisa dipakai oleh Kaprodi

5. Lihat data dosen

Nama Fungsi	Lihat data dosen
Kondisi awal	Aktor telah terotentikasi
Alur kerja	<ol style="list-style-type: none"> 1. Klik menu Dosen 2. Sistem menampilkan daftar dosen Jurdikmat UNY 3. Klik detail 4. Sistem menampilkan detail data dosen
Kondisi akhir	Detail informasi dosen
Pengecualian	-
Kebutuhan non-fungsional	Fungsi ini bisa dipakai oleh semua <i>user</i> yang <i>login</i>

6. Ubah data dosen

Nama Fungsi	Ubah data dosen
Kondisi awal	Aktor telah terotentikasi sebagai Staf
Alur kerja	<ol style="list-style-type: none"> 1. Klik menu Dosen 2. Sistem menampilkan daftar dosen Jurdikmat UNY 3. Klik edit 4. Sistem menampilkan form data dosen 5. Ubah data 6. Simpan perubahan
Kondisi akhir	Data dosen berubah
Pengecualian	-
Kebutuhan non-fungsional	Fungsi ini hanya bisa dipakai oleh Staf

7. Hak akses

Nama Fungsi	Hak akses
Kondisi awal	Aktor telah terotentikasi sebagai Staf
Alur kerja	<ol style="list-style-type: none"> 1. Klik menu Hak akses 2. Sistem menampilkan daftar pengguna 3. Klik NIP/NIM pengguna 4. Pilih Hak akses pengguna tersebut 5. Simpan data
Kondisi akhir	Hak akses pengguna telah ditentukan
Pengecualian	-
Kebutuhan non-fungsional	Fungsi ini hanya bisa dipakai oleh Staf

8. Pesan

Nama Fungsi	Pesan
Kondisi awal	Aktor telah terotentikasi
Alur kerja	<p>Klik menu Pesan</p> <ol style="list-style-type: none"> a. Pesan masuk <ol style="list-style-type: none"> 1. Klik pesan masuk 2. Sistem menampilkan pesan yang diterima 3. Klik subjek pesan 4. Sistem menampilkan isi pesan dan percakapan 5. Isi <i>field</i> balas pesan (jika ada) 6. Klik balas pesan (jika poin 5 dilakukan) b. Buat pesan <ol style="list-style-type: none"> 1. Klik buat pesan

	2. Isi tujuan pengiriman, subjek, dan isi pesan 3. Klik kirim pesan
Kondisi akhir	a. Sistem menampilkan isi pesan b. Sistem mengirim pesan
Pengecualian	-
Kebutuhan non-fungsional	Fungsi ini bisa dipakai oleh semua <i>user</i> yang <i>login</i>

9. Data pengguna

Nama Fungsi	Data pengguna
Kondisi awal	Aktor telah terotentikasi sebagai Staf
Alur kerja	1. Klik menu User 2. Sistem menampilkan daftar pengguna 3. Klik edit 4. Sistem menampilkan form pengguna 5. Ubah data 6. Simpan data
Kondisi akhir	Data pengguna telah diubah
Pengecualian	Semua pengguna dapat mengubah data pribadi (selain NIM/NIP dan status keaktifan) melalui profil
Kebutuhan non-fungsional	Fungsi ini hanya bisa dipakai oleh Staf

10. Profil

Nama Fungsi	Profil
Kondisi awal	Aktor telah terotentikasi
Alur kerja	1. Klik menu Profil 2. Sistem menampilkan profil pengguna 3. Klik edit 4. Sistem menampilkan detail data dosen
Kondisi akhir	Detail informasi dosen
Pengecualian	-
Kebutuhan non-fungsional	Fungsi ini bisa dipakai oleh semua <i>user</i> yang <i>login</i>

3.6 Kebutuhan Data

Data yang digunakan dalam aplikasi bimbingan skripsi *online* berdasarkan dari prosedur yang dilakukan. Data-data tersebut adalah sebagai berikut.

1. Draft skripsi

Data yang memuat karya tulis mahasiswa, judul skripsi, dan data mahasiswa (NIM, nama). Sumber data ini adalah Mahasiswa.

2. Arsip judul skripsi

Data yang memuat informasi tentang judul skripsi yang sudah digunakan/dipakai. Sumber data ini adalah Staf dan Kaprodi.

3. Data dosen

Data yang memuat profil dosen Jurdikmat UNY. Sumber data ini adalah Staf.

3.7 Kebutuhan Non-fungsional

Berikut adalah beberapa kebutuhan non-fungsional yang dibutuhkan Aplikasi Bimbingan Skripsi *Online*.

1. Ketersediaan

- a. Sistem harus tersedia dalam kurun waktu 1 x 24 jam.
- b. Sistem harus memiliki penyimpanan yang cukup untuk menampung data.

2. Performa

Sistem harus dapat memberikan layanan secara cepat bagi penggunaanya (proses tidak melebihi 1 menit).

3. Keamanan

Sistem harus mempunyai kategori user login (Administrator, Dosen, Kaprodi, Mahasiswa, dan Staf).

KODE PROGRAM

1. Login

a. Model

UserLogin.php

```
<?php
```

```
/**
 * LoginForm class.
 * LoginForm is the data structure for keeping
 * user login form data. It is used by the 'login' action of
 * 'SiteController'.
 */
class UserLogin extends CFormModel
{
    public $username;
    public $password;
    public $rememberMe;

    /**
     * Declares the validation rules.
     * The rules state that username and password are required,
     * and password needs to be authenticated.
     */
    public function rules()
    {
        return array(
            // username and password are required
            array('username, password', 'required'),
            // rememberMe needs to be a boolean
            array('rememberMe', 'boolean'),
            // password needs to be authenticated
            array('password', 'authenticate'),
        );
    }

    /**
     * Declares attribute labels.
     */
    public function attributeLabels()
    {
        return array(
            'rememberMe'=>UserModule::t("Remember me next
time"),
            'username'=>UserModule::t("username or email"),
            'password'=>UserModule::t("password"),
        );
    }

    /**
     * Authenticates the password.
     * This is the 'authenticate' validator as declared in
rules().
     */
    public function authenticate($attribute,$params)
    {
        if(!$this->hasErrors()) // we only want to
authenticate when no input errors

```

```

        {
            $identity=new UserIdentity($this->username,$this->password);
            $identity->authenticate();
            switch($identity->errorCode)
            {
                case UserIdentity::ERROR_NONE:
                    $duration=$this->rememberMe ?
Yii::app()->controller->module->rememberMeTime : 0;
                    Yii::app()->user->login($identity,$duration);
                    break;
                case UserIdentity::ERROR_EMAIL_INVALID:
                    $this->addError("username",UserModule::t("Email is incorrect."));
                    break;
                case UserIdentity::ERROR_USERNAME_INVALID:
                    $this->addError("username",UserModule::t("Username is incorrect."));
                    break;
                case UserIdentity::ERROR_STATUS_NOTACTIV:
                    $this->addError("status",UserModule::t("You account is not
activated."));
                    break;
                case UserIdentity::ERROR_STATUS_BAN:
                    $this->addError("status",UserModule::t("You account is blocked."));
                    break;
                case UserIdentity::ERROR_PASSWORD_INVALID:
                    $this->addError("password",UserModule::t("Password is incorrect."));
                    break;
            }
        }
    }
}

```

b. View

login.php

```

<h1><?php echo UserModule::t("Login"); ?></h1>

<?php if(Yii::app()->user->hasFlash('loginMessage')): ?>

<div class="success">
    <?php echo Yii::app()->user->getFlash('loginMessage'); ?>
</div>

<?php endif; ?>

<div class="form">
<?php echo CHtml::beginForm(); ?>

    <p class="note"><?php echo UserModule::t('Isi NIM/NIP tanpa
spasi'); ?></p>

```

```

        <?php echo CHtml::errorSummary($model); ?>

        <div class="row">
            <?php echo CHtml::activeLabelEx($model, 'NIM/NIP '); ?>
            <?php echo CHtml::activeTextField($model, 'username')
?>
        </div>

        <div class="row">
            <?php echo CHtml::activeLabelEx($model, 'Password ');
?>
            <?php echo
CHtml::activePasswordField($model, 'password') ?>
        </div>

        <div class="row submit">
            <?php echo
CHtml::submitButton(UserModule::t("Login")); ?>
        </div>

        <div class="row">
            <p class="hint">
                <?php //echo CHtml::link(UserModule::t("Lupa
Password?"),Yii::app()->getModule('user')->recoveryUrl); ?>
                <?php echo
CHtml::link(UserModule::t("Daftar"),Yii::app()->getModule('user')-
>registrationUrl); ?>
            </p>
        </div>

        <div class="row rememberMe">
            <?php /*echo
CHtml::activeCheckBox($model, 'rememberMe'); ?>
            <?php echo CHtml::activeLabelEx($model, 'rememberMe');
*/?>
        </div>

<?php echo CHtml::endForm(); ?>
</div><!-- form -->

<?php
$form = new CForm(array(
    'elements'=>array(
        'username'=>array(
            'type'=>'text',
            'maxlength'=>32,
        ),
        'password'=>array(
            'type'=>'password',
            'maxlength'=>32,
        ),
        'rememberMe'=>array(
            'type'=>'checkbox',
        )
    ),
),

```

```

        'buttons'=>array(
            'login'=>array(
                'type'=>'submit',
                'label'=>'Login',
            ),
        ),
    ), $model);
?>

```

c. Controller LoginController.php

```

<?php

class LoginController extends Controller
{
    public $defaultAction = 'login';

    /**
     * Displays the login page
     */
    public function actionLogin()
    {
        if (Yii::app()->user->isGuest) {
            $model=new UserLogin;
            // collect user input data
            if(isset($_POST['UserLogin']))
            {
                $model->attributes=$_POST['UserLogin'];
                // validate user input and redirect to
previous page if valid
                if($model->validate()) {
                    $this->lastViset();
                    if (Yii::app()->user-
>returnUrl=='/index.php')
                        $this->redirect(Yii::app()-
>controller->module->returnUrl);
                    else
                        $this->redirect(Yii::app()-
>user->returnUrl);
                }
            }
            // display the login form
            $this-
>render('/user/login',array('model'=>$model));
        } else
            $this->redirect(Yii::app()->controller->module-
>returnUrl);
    }

    private function lastViset() {
        $lastVisit = User::model()->notsafe()-
>findByPk(Yii::app()->user->id);
        $lastVisit->lastvisit = time();
        $lastVisit->save();
    }
}

```

```
}
```

2. Registrasi

a. Model

RegistrationForm.php

```
<?php
/**
 * RegistrationForm class.
 * RegistrationForm is the data structure for keeping
 * user registration form data. It is used by the 'registration'
action of 'UserController'.
 */
class RegistrationForm extends User {
    public $verifyPassword;
    public $verifyCode;

    public function rules() {
        $rules = array(
            array('username, password, verifyPassword,
email', 'required'),
            array('username', 'length', 'max'=>20, 'min' =>
3, 'message' => UserModule::t("Incorrect username (length between 3
and 20 characters).")),
            array('password', 'length', 'max'=>128, 'min' =>
4, 'message' => UserModule::t("Incorrect password (minimal length 4
symbols).")),
            array('email', 'email'),
            array('username', 'unique', 'message' =>
UserModule::t("This user's name already exists.")),
            array('email', 'unique', 'message' =>
UserModule::t("This user's email address already exists.")),
            //array('verifyPassword', 'compare',
'compareAttribute'=>'password', 'message' => UserModule::t("Retype
Password is incorrect.")),
            array('username', 'match', 'pattern' => '/^[A-
Za-z0-9_]+$/', 'message' => UserModule::t("Incorrect symbols (A-
z0-9).")),
        );
        if (!(isset($_POST['ajax']) &&
$_POST['ajax']=='registration-form')) {
            array_push($rules, array('verifyCode', 'captcha',
'allowEmpty'=>!UserModule::doCaptcha('registration')));
        }

        array_push($rules, array('verifyPassword', 'compare',
'compareAttribute'=>'password', 'message' => UserModule::t("Retype
Password is incorrect.")));
        return $rules;
    }
}
```

b. View

registration.php

```
<?php $this->pageTitle=Yii::app()->name . ' -
'.UserModule::t("Registration");
$this->breadcrumbs=array(
    UserModule::t("Registration"),
);
?>

<h1><?php echo UserModule::t("Registration"); ?></h1>

<?php if(Yii::app()->user->hasFlash('registration')): ?>
<div class="success">
<?php echo Yii::app()->user->getFlash('registration'); ?>
</div>
<?php else: ?>

<div class="form">
<?php $form=$this->beginWidget('UActiveForm', array(
    'id'=>'registration-form',
    'enableAjaxValidation'=>true,
    'disableAjaxValidationAttributes'=>array('RegistrationForm_v
erifyCode'),
    'clientOptions'=>array(
        'validateOnSubmit'=>true,
    ),
    'htmlOptions' => array('enctype'=>'multipart/form-data'),
)); ?>

    <p class="note"><?php echo UserModule::t('Fields with <span
class="required">*</span> are required.');
```

```

.    <div class="row">
        <?php echo $form->labelEx($model,'email'); ?>
        <?php echo $form->textField($model,'email'); ?>
        <?php echo $form->error($model,'email'); ?>
    </div>

<?php
    $profileFields=$profile->getFields();
    if ($profileFields) {
        foreach($profileFields as $field) {
            ?>
            <div class="row">
                <?php echo $form->labelEx($profile,$field->varname);
            ?>

            <?php
                if ($widgetEdit = $field->widgetEdit($profile)) {
                    echo $widgetEdit;
                } elseif ($field->range) {
                    echo $form->dropDownList($profile,$field-
>varname,Profile::range($field->range));
                } elseif ($field->field_type=="TEXT") {
                    echo $form->textArea($profile,$field-
>varname,array('rows'=>6, 'cols'=>50));
                } else {
                    echo $form->textField($profile,$field-
>varname,array('size'=>60, 'maxlength'=>(($field-
>field_size)?$field->field_size:255)));
                }
            ?>
            <?php echo $form->error($profile,$field->varname); ?>
            </div>

            <?php
                }
            ?>

            <?php if (UserModule::doCaptcha('registration')): ?>
            <div class="row">
                <?php echo $form->labelEx($model,'verifyCode'); ?>

                <?php $this->widget('CCaptcha'); ?>
                <?php echo $form->textField($model,'verifyCode'); ?>
                <?php echo $form->error($model,'verifyCode'); ?>

                <p class="hint"><?php echo UserModule::t("Please enter
the letters as they are shown in the image above."); ?>
                <br/><?php echo UserModule::t("Letters are not case-
sensitive."); ?></p>
            </div>
            <?php endif; ?>

            <div class="row submit">
                <?php echo
CHtml::submitButton(UserModule::t("Register")); ?>
            </div>

            <?php $this->endWidget(); ?>

```



```
</div><!-- form -->
<?php endif; ?>
```

c. Controller

RegistrationController.php

```
<?php
```

```
class RegistrationController extends Controller
{
    public $defaultAction = 'registration';

    /**
     * Declares class-based actions.
     */
    public function actions()
    {
        return array(
            'captcha'=>array(
                'class'=>'CCaptchaAction',
                'backColor'=>0xFFFFFF,
            ),
        );
    }
    /**
     * Registration user
     */
    public function actionRegistration() {
        $model = new RegistrationForm;
        $profile=new Profile;
        $profile->regMode = true;

        // ajax validator
        if(isset($_POST['ajax']) &&
        $_POST['ajax']==='registration-form')
        {
            echo
        UActiveForm::validate(array($model,$profile));
            Yii::app()->end();
        }

        if (Yii::app()->user->id) {
            $this->redirect(Yii::app()->controller->module->profileUrl);
        } else {
            if(isset($_POST['RegistrationForm'])) {
                $model->attributes=$_POST['RegistrationForm'];
                $profile->attributes=(isset($_POST['Profile'])?$_POST['Profile']:array());
                ;
                if($model->validate()&&$profile->validate())
                {
                    $sourcePassword = $model->password;
                }
            }
        }
    }
}
```

```

        $model->
>activkey=UserModule::encrypting(microtime().$model->password);
        $model->
>password=UserModule::encrypting($model->password);
        $model->
>verifyPassword=UserModule::encrypting($model->verifyPassword);
        $model->superuser=0;
        $model->status=((Yii::app()-
>controller->module->
>activeAfterRegister)?User::STATUS_ACTIVE:User::STATUS_NOACTIVE);

        if ($model->save()) {
            $profile->
>user_id=$model->id;
            $profile->save();
            if (Yii::app()-
>controller->module->sendActivationMail) {
                $activation_url =
$this->
>createAbsoluteUrl('/user/activation/activation',array("activkey"
=> $model->activkey, "email" => $model->email));

                UserModule::sendMail($model->email,UserModule::t("You
registered from {site_name}",array('{site_name}'=>Yii::app()-
>name)),UserModule::t("Please activate you account go to
{activation_url}",array('{activation_url}'=>$activation_url)));
            }

            if ((Yii::app()-
>controller->module->loginNotActiv|| (Yii::app()->controller->
>module->activeAfterRegister&&Yii::app()->controller->module->
>sendActivationMail==false))&&Yii::app()->controller->module->
>autoLogin) {

                $identity=new UserIdentity($model->username,$soucePassword);
                $identity->
>authenticate();
                Yii::app()-
>user->login($identity,0);
                $this->
>redirect(Yii::app()->controller->module->returnUrl);
            } else {
                if (!Yii::app()-
>controller->module->activeAfterRegister&&!Yii::app()->controller->
>module->sendActivationMail) {
                    Yii::app()-
>user->setFlash('registration',UserModule::t("Thank you for your
registration. Contact Administrator to activate your account."));
                }

                elseif(Yii::app()->controller->module->
>activeAfterRegister&&Yii::app()->controller->module->
>sendActivationMail==false) {
                    Yii::app()-
>user->setFlash('registration',UserModule::t("Thank you for your
registration. Please

```

```

{{login}}.",array('{{login}}'=>CHtml::link(UserModule::t('Login'),
Yii::app()->controller->module->loginUrl))));
    }
elseif(Yii::app()->controller->module->loginNotActiv) {
    Yii::app()-
>user->setFlash('registration',UserModule::t("Thank you for your
registration. Please check your email or login."));
    } else {
    Yii::app()-
>user->setFlash('registration',UserModule::t("Thank you for your
registration. Please check your email."));
    }
    $this->refresh();
}
}
} else $profile->validate();
}
$this->
>render('/user/registration',array('model'=>$model,'profile'=>$pro
file));
}
}
}

```

3. Judul

a. Model

Mahasiswa.php

<?php

```

/**
 * This is the model class for table "mahasiswa".
 *
 * The followings are the available columns in table 'mahasiswa':
 * @property integer $id_mhs
 * @property string $nim_mhs
 * @property string $nama_mhs
 * @property string $fokus_studi
 * @property string $judul_skripsi
 * @property string $pembimbing
 * @property string $tgl_masuk
 * @property string $tgl_ujian
 * @property string $penguji1
 * @property string $penguji2
 */
class Mahasiswa extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'mahasiswa';
    }

    /**

```

```

        * @return array validation rules for model attributes.
        */
        public function rules()
        {
            // NOTE: you should only define rules for those
attributes that
            // will receive user inputs.
            return array(
                array('nim_mhs, nama_mhs, fokus_studi,
judul_skripsi, pembimbing, tgl_masuk', 'required'),
                array('nim_mhs', 'length', 'max'=>11),
                array('nama_mhs, pembimbing, pengujil,
penguji2', 'length', 'max'=>50),
                array('fokus_studi', 'length', 'max'=>10),
                array('judul_skripsi', 'length', 'max'=>250),
                array('tgl_ujian', 'safe'),
                // The following rule is used by search().
                // @todo Please remove those attributes that
should not be searched.
                array('id_mhs, nim_mhs, nama_mhs, fokus_studi,
judul_skripsi, pembimbing, tgl_masuk, tgl_ujian, pengujil,
penguji2', 'safe', 'on'=>'search'),
            );
        }

        /**
        * @return array relational rules.
        */
        public function relations()
        {
            // NOTE: you may need to adjust the relation name and
the related
            // class name for the relations automatically
generated below.
            return array(
            );
        }

        /**
        * @return array customized attribute labels (name=>label)
        */
        public function attributeLabels()
        {
            return array(
                'id_mhs' => 'Id Mahasiswa',
                'nim_mhs' => 'NIM',
                'nama_mhs' => 'Nama Mahasiswa',
                'fokus_studi' => 'Fokus Studi',
                'judul_skripsi' => 'Judul Skripsi',
                'pembimbing' => 'Pembimbing',
                'tgl_masuk' => 'Tanggal Masuk',
                'tgl_ujian' => 'Tanggal Ujian',
                'pengujil' => 'Penguji 1',
                'penguji2' => 'Penguji 2',
            );
        }

```

```

    /**
     * Retrieves a list of models based on the current
search/filter conditions.
     *
     * Typical usecase:
     * - Initialize the model fields with values from filter
form.
     * - Execute this method to get CActiveDataProvider instance
which will filter
     * models according to data in model fields.
     * - Pass data provider to CGridView, CListView or any
similar widget.
     *
     * @return CActiveDataProvider the data provider that can
return the models
     * based on the search/filter conditions.
     */
    public function search()
    {
        // @todo Please modify the following code to remove
attributes that should not be searched.

        $criteria=new CDbCriteria;

        $criteria->compare('id_mhs',$this->id_mhs);
        $criteria->compare('nim_mhs',$this->nim_mhs,true);
        $criteria->compare('nama_mhs',$this->nama_mhs,true);
        $criteria->compare('fokus_studi',$this->fokus_studi,true);
        $criteria->compare('judul_skripsi',$this->judul_skripsi,true);
        $criteria->compare('pembimbing',$this->pembimbing,true);
        $criteria->compare('tgl_masuk',$this->tgl_masuk,true);
        $criteria->compare('tgl_ujian',$this->tgl_ujian,true);
        $criteria->compare('pengujil',$this->pengujil,true);
        $criteria->compare('penguji2',$this->penguji2,true);

        return new CActiveDataProvider($this, array(
            'criteria'=>$criteria,
        ));
    }

    /**
     * Returns the static model of the specified AR class.
     * Please note that you should have this exact method in all
your CActiveRecord descendants!
     * @param string $className active record class name.
     * @return Mahasiswa the static model class
     */
    public static function model($className=__CLASS__)
    {
        return parent::model($className);
    }
}

```

b. View

admin.php (Daftar Judul)

```
<?php
/* @var $this MahasiswaController */
/* @var $model Mahasiswa */

$this->breadcrumbs=array(
    'Mahasiswas'=>array('index'),
    'Manage',
);

$this->menu=array(
    //array('label'=>'List Mahasiswa', 'url'=>array('index')),
    array('label'=>'Tambah Daftar',
    'url'=>array('create'), 'visible'=>Yii::app()->user-
>checkAccess('Staff')),
    array('label'=>'Tambah Daftar',
    'url'=>array('create'), 'visible'=>Yii::app()->user-
>checkAccess('Kaprodi')),
);

Yii::app()->clientScript->registerScript('search', "
$('.search-button').click(function(){
    $('.search-form').toggle();
    return false;
});
$('.search-form form').submit(function(){
    $('#mahasiswa-grid').yiiGridView('update', {
        data: $(this).serialize()
    });
    return false;
});
");
?>

<h1>Daftar Judul Skripsi</h1>

<?php $this->widget('application.extensions.tablesorter.Sorter',
array(
    'data'=>$records,
    'columns'=>array(
        'nim_mhs',
        'nama_mhs',
        'judul_skripsi',
        'tgl_masuk',
        'tgl_ujian',
        /*
        'id_mhs',
        'fokus_studi',
        'pembimbing',
        'penguji1',
        'penguji2',
        */
    ),
)); ?>
```

create.php

```
<?php
/* @var $this MahasiswaController */
/* @var $model Mahasiswa */

$this->breadcrumbs=array(
    'Mahasiswas'=>array('index'),
    'Create',
);

$this->menu=array(
    //array('label'=>'List Mahasiswa', 'url'=>array('index')),
    array('label'=>'Daftar Skripsi', 'url'=>array('admin')),
);
?>

<h1>Tambah Judul Skripsi</h1>

<?php $this->renderPartial('_form', array('model'=>$model)); ?>
```

_form.php

```
<?php
/* @var $this MahasiswaController */
/* @var $model Mahasiswa */
/* @var $form CActiveForm */
?>

<div class="form">

<?php $form=$this->beginWidget('CActiveForm', array(
    'id'=>'mahasiswa-form',
    // Please note: When you enable ajax validation, make sure
the corresponding
    // controller action is handling ajax validation correctly.
    // There is a call to performAjaxValidation() commented in
generated controller code.
    // See class documentation of CActiveForm for details on
this.
    'enableAjaxValidation'=>false,
)); ?>

    <p class="note">Fields dengan tanda <span
class="required">*</span> harus diisi.</p>

    <?php echo $form->errorSummary($model); ?>

    <div class="row">
        <?php echo $form->labelEx($model,'nim_mhs'); ?>
        <?php echo $form-
>textField($model,'nim_mhs',array('size'=>11,'maxlength'=>11)); ?>
        <?php echo $form->error($model,'nim_mhs'); ?>
    </div>

    <div class="row">
        <?php echo $form->labelEx($model,'nama_mhs'); ?>
```

```

        <?php echo $form-
>textField($model,'nama_mhs',array('size'=>50,'maxlength'=>50));
?>

        <?php echo $form->error($model,'nama_mhs'); ?>
    </div>

    <div class="row">
        <?php echo $form->labelEx($model,'fokus_studi'); ?>
        <?php echo $form-
>dropDownList($model,'fokus_studi',array('aljabar'=>'Aljabar','geo
metri'=>'Geometri','komputer'=>'Komputer','statistika'=>'Statistik
a','terapan'=>'Terapan','pendidikan'=>'Pendidikan')); ?>
        <?php echo $form->error($model,'fokus_studi'); ?>
    </div>

    <div class="row">
        <?php echo $form->labelEx($model,'judul_skripsi'); ?>
        <?php echo $form-
>textField($model,'judul_skripsi',array('size'=>60,'maxlength'=>25
0)); ?>
        <?php echo $form->error($model,'judul_skripsi'); ?>
    </div>

    <div class="row">
        <?php echo $form->labelEx($model,'pembimbing'); ?>
        <?php echo $form-
>textField($model,'pembimbing',array('size'=>50,'maxlength'=>50));
?>
        <?php echo $form->error($model,'pembimbing'); ?>
    </div>

    <div class="row">
        <?php echo $form->labelEx($model,'tgl_masuk'); ?>
        <?php $this->widget('zii.widgets.jui.CJuiDatePicker',
            array(
                // you must specify name or
model/attribute
                'model'=>$model,
                'attribute'=>'tgl_masuk',
                'value' => $model->tgl_masuk,
                'options' => array(
                    'dateFormat'=>'yy-mm-dd',
                    'showOn'=>'button',
                    'yearRange'=>'-60',
                    'changeMonth' => 'true',
                    'changeYear' => 'true',
                    'constrainInput' => 'false',
                    'duration'=>'fast',
                    'showAnim' =>'slide',
                    //'showAnim' =>'fold',
                ),
                'htmlOptions'=>array('size'=>20,)
            )
        ); ?>
        <?php echo $form->error($model,'tgl_masuk'); ?>
    </div>

```



```

        <div class="row">
            <?php echo $form->labelEx($model,'tgl_ujian'); ?>
            <?php $this->widget('zii.widgets.jui.CJuiDatePicker',
                array(
                    // you must specify name or
model/attribute
                    'model'=>$model,
                    'attribute'=>'tgl_ujian',
                    'value' => $model->tgl_ujian,
                    'options' => array(
                        'dateFormat'=>'yy-mm-dd',
                        'showOn'=>'button',
                        'yearRange'=>'-60',
                        'changeMonth' => 'true',
                        'changeYear' => 'true',
                        'constrainInput' => 'false',
                        'duration'=>'fast',
                        'showAnim' =>'slide',
                        //'showAnim' =>'fold',
                    ),
                    'htmlOptions'=>array('size'=>20,),
                )
            ); ?>
            <?php echo $form->error($model,'tgl_ujian'); ?>
        </div>

        <div class="row">
            <?php echo $form->labelEx($model,'penguji1'); ?>
            <?php echo $form-
>textField($model,'penguji1',array('size'=>50,'maxlength'=>50));
?>
            <?php echo $form->error($model,'penguji1'); ?>
        </div>

        <div class="row">
            <?php echo $form->labelEx($model,'penguji2'); ?>
            <?php echo $form-
>textField($model,'penguji2',array('size'=>50,'maxlength'=>50));
?>
            <?php echo $form->error($model,'penguji2'); ?>
        </div>

        <div class="row buttons">
            <?php echo CHtml::submitButton($model->isNewRecord ?
'Tambah' : 'Save'); ?>
        </div>

        <?php $this->endWidget(); ?>

    </div><!-- form -->

```

view.php

```

<?php
/* @var $this MahasiswaController */
/* @var $model Mahasiswa */

```

```

$this->breadcrumbs=array(
    'Mahasiswas'=>array('index'),
    $model->id_mhs,
);

$this->menu=array(
    //array('label'=>'List Mahasiswa', 'url'=>array('index')),
    array('label'=>'Tambah Daftar',
        'url'=>array('create'), 'visible'=>Yii::app()->user-
        >checkAccess('Staff')),
    array('label'=>'Tambah Daftar',
        'url'=>array('create'), 'visible'=>Yii::app()->user-
        >checkAccess('Kaprodi')),
    array('label'=>'Update Data', 'url'=>array('update',
        'id'=>$model->id_mhs), 'visible'=>Yii::app()->user-
        >checkAccess('Staff')),
    array('label'=>'Update Data', 'url'=>array('update',
        'id'=>$model->id_mhs), 'visible'=>Yii::app()->user-
        >checkAccess('Kaprodi')),
    array('label'=>'Hapus Data', 'url'=>'#',
        'linkOptions'=>array('submit'=>array('delete', 'id'=>$model-
        >id_mhs), 'confirm'=>'Anda yakin?'), 'visible'=>Yii::app()->user-
        >checkAccess('Staff')),
    array('label'=>'Hapus Data', 'url'=>'#',
        'linkOptions'=>array('submit'=>array('delete', 'id'=>$model-
        >id_mhs), 'confirm'=>'Anda yakin?'), 'visible'=>Yii::app()->user-
        >checkAccess('Kaprodi')),
    array('label'=>'Daftar Judul Skripsi',
        'url'=>array('admin')),
);
?>

```

```

<h1>Detail Judul #<?php echo $model->nim_mhs; ?></h1>

```

```

<?php $this->widget('zii.widgets.CDetailView', array(
    'data'=>$model,
    'attributes'=>array(
        'nim_mhs',
        'nama_mhs',
        'fokus_studi',
        'judul_skripsi',
        'pembimbing',
        'tgl_masuk',
        'tgl_ujian',
        'pengujil',
        'penguji2',
    ),
)); ?>

```

_view.php

```

<?php
/* @var $this MahasiswaController */
/* @var $data Mahasiswa */
?>

```

```

<div class="view">

    <b><?php echo CHtml::encode($data-
>getAttributeLabel('nim_mhs')); ?>:</b>
    <?php echo CHtml::link(CHtml::encode($data->nim_mhs),
array('view', 'id'=>$data->id_mhs)); ?>
    <br />

    <b><?php echo CHtml::encode($data-
>getAttributeLabel('nama_mhs')); ?>:</b>
    <?php echo CHtml::encode($data->nama_mhs); ?>
    <br />

    <b><?php echo CHtml::encode($data-
>getAttributeLabel('fokus_studi')); ?>:</b>
    <?php echo CHtml::encode($data->fokus_studi); ?>
    <br />

    <b><?php echo CHtml::encode($data-
>getAttributeLabel('judul_skripsi')); ?>:</b>
    <?php echo CHtml::encode($data->judul_skripsi); ?>
    <br />

    <b><?php echo CHtml::encode($data-
>getAttributeLabel('pembimbing')); ?>:</b>
    <?php echo CHtml::encode($data->pembimbing); ?>
    <br />

    <b><?php echo CHtml::encode($data-
>getAttributeLabel('tgl_masuk')); ?>:</b>
    <?php echo CHtml::encode($data->tgl_masuk); ?>
    <br />

    <?php /*
    <b><?php echo CHtml::encode($data-
>getAttributeLabel('tgl_ujian')); ?>:</b>
    <?php echo CHtml::encode($data->tgl_ujian); ?>
    <br />

    <b><?php echo CHtml::encode($data-
>getAttributeLabel('penguji1')); ?>:</b>
    <?php echo CHtml::encode($data->penguji1); ?>
    <br />

    <b><?php echo CHtml::encode($data-
>getAttributeLabel('penguji2')); ?>:</b>
    <?php echo CHtml::encode($data->penguji2); ?>
    <br />

    */ ?>

</div>

```

c. Controller

MahasiswaController.php

<?php

```
class MahasiswaController extends Controller
{
    /**
     * @var string the default layout for the views. Defaults to
     * '//layouts/column2', meaning
     * using two-column layout. See
     * 'protected/views/layouts/column2.php'.
     */
    public $layout='//layouts/column2';

    /**
     * @return array action filters
     */
    public function filters()
    {
        return array(
            // 'rights',
            'accessControl', // perform access control for
            CRUD operations
            'postOnly + delete', // we only allow deletion
            via POST request
        );
    }

    /**
     * Specifies the access control rules.
     * This method is used by the 'accessControl' filter.
     * @return array access control rules
     */
    public function accessRules()
    {
        return array(
            array('allow', // allow authenticated user to
            perform 'create' and 'update' actions
                'actions'=>array('admin','view','index'),
                'users'=>array('@'),
            ),
            array('allow', // allow admin user to perform
            'admin' and 'delete' actions
                'actions'=>array('create','update','delete'),
                'hak
            akses'=>array('SuperAdmin','Staff','Kaprodi'),
            ),
            array('deny', // deny all users
                'users'=>array('*'),
            ),
        );
    }

    /**
```

```

    * Displays a particular model.
    * @param integer $id the ID of the model to be displayed
    */
    public function actionView($id)
    {
        $this->render('view',array(
            'model'=>$this->loadModel($id),
        ));
    }

    /**
     * Creates a new model.
     * If creation is successful, the browser will be redirected
to the 'view' page.
    */
    public function createAction()
    {
        $model=new Mahasiswa;

        // Uncomment the following line if AJAX validation is
needed
        // $this->performAjaxValidation($model);

        if(isset($_POST['Mahasiswa']))
        {
            $model->attributes=$_POST['Mahasiswa'];
            if($model->save())
                $this->redirect(array('view','id'=>$model-
>id_mhs));
        }

        $this->render('create',array(
            'model'=>$model,
        ));
    }

    /**
     * Updates a particular model.
     * If update is successful, the browser will be redirected
to the 'view' page.
     * @param integer $id the ID of the model to be updated
     */
    public function actionUpdate($id)
    {
        $model=$this->loadModel($id);

        // Uncomment the following line if AJAX validation is
needed
        // $this->performAjaxValidation($model);

        if(isset($_POST['Mahasiswa']))
        {
            $model->attributes=$_POST['Mahasiswa'];
            if($model->save())
                $this->redirect(array('view','id'=>$model-
>id_mhs));
        }
    }

```

```

        }

        $this->render('update',array(
            'model'=>$model,
        ));
    }

    /**
     * Deletes a particular model.
     * If deletion is successful, the browser will be redirected
to the 'admin' page.
     * @param integer $id the ID of the model to be deleted
     */
    public function actionDelete($id)
    {
        $this->loadModel($id)->delete();

        // if AJAX request (triggered by deletion via admin
grid view), we should not redirect the browser
        if(!isset($_GET['ajax']))
            $this->redirect(isset($_POST['returnUrl']) ?
$_POST['returnUrl'] : array('admin'));
    }

    /**
     * Lists all models.
     */
    public function actionIndex()
    {
        $dataProvider=new CActiveDataProvider('Mahasiswa');
        $this->render('index',array(
            'dataProvider'=>$dataProvider,
        ));
    }

    /**
     * Manages all models.
     */
    public function actionAdmin ()
    {
        $records=Mahasiswa::model()->findAll();
        $this->render('admin',array(
            'records'=>$records,
        ));
    }

    /**
     * Returns the data model based on the primary key given in
the GET variable.
     * If the data model is not found, an HTTP exception will be
raised.
     * @param integer $id the ID of the model to be loaded
     * @return Mahasiswa the loaded model
     * @throws CHttpException
     */
    public function loadModel($id)

```

```

    {
        $model=Mahasiswa::model()->findByPk($id);
        if($model===null)
            throw new CHttpException(404,'The requested page
does not exist.');
```

return \$model;

```

    }

    /**
     * Performs the AJAX validation.
     * @param Mahasiswa $model the model to be validated
     */
    protected function performAjaxValidation($model)
    {
        if(isset($_POST['ajax']) &&
$_POST['ajax']==='mahasiswa-form')
        {
            echo CActiveForm::validate($model);
            Yii::app()->end();
        }
    }
}

```

4. Proposal

a. Model

```
<?php
```

```

/**
 * This is the model class for table "proposal".
 *
 * The followings are the available columns in table 'proposal':
 * @property integer $id_proposal
 * @property string $nim_mhs
 * @property string $judul_proposal
 * @property string $file_proposal
 */
class Proposal extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'proposal';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those
attributes that
        // will receive user inputs.
        return array(
            array('nim_mhs, judul_proposal', 'required'),

```

```

        array('nim_mhs', 'length', 'max'=>20),
        array('judul_proposal', 'length', 'max'=>250),
        array('file_proposal', 'file', 'types'=>'doc,
docx, pdf'),
        // The following rule is used by search().
        // @todo Please remove those attributes that
should not be searched.
        array('id_proposal, nim_mhs, judul_proposal,
file_proposal', 'safe', 'on'=>'search'),
    );
}

/**
 * @return array relational rules.
 */
public function relations()
{
    // NOTE: you may need to adjust the relation name and
the related
    // class name for the relations automatically
generated below.
    return array(
    );
}

/**
 * @return array customized attribute labels (name=>label)
 */
public function attributeLabels()
{
    return array(
        'id_proposal' => 'Id Proposal',
        'nim_mhs' => 'Nim Mahasiswa',
        'judul_proposal' => 'Judul Proposal',
        'file_proposal' => 'File Proposal',
    );
}

/**
 * Retrieves a list of models based on the current
search/filter conditions.
 *
 * Typical usecase:
 * - Initialize the model fields with values from filter
form.
 * - Execute this method to get CActiveDataProvider instance
which will filter
 * models according to data in model fields.
 * - Pass data provider to CGridView, CListView or any
similar widget.
 *
 * @return CActiveDataProvider the data provider that can
return the models
 * based on the search/filter conditions.
 */
public function search()

```



```

        {
            // @todo Please modify the following code to remove
            attributes that should not be searched.

            $criteria=new CDbCriteria;

            $criteria->compare('id_proposal',$this->id_proposal);
            $criteria->compare('nim_mhs',$this->nim_mhs,true);
            $criteria->compare('judul_proposal',$this->
>judul_proposal,true);
            $criteria->compare('file_proposal',$this->
>file_proposal,true);

            return new CActiveDataProvider($this, array(
                'criteria'=>$criteria,
            ));
        }

        /**
         * Returns the static model of the specified AR class.
         * Please note that you should have this exact method in all
your CActiveRecord descendants!
         * @param string $className active record class name.
         * @return Proposal the static model class
         */
        public static function model($className=__CLASS__)
        {
            return parent::model($className);
        }
    }
}

```

b. View

admin.php (List Proposal)

```

<?php
/* @var $this ProposalController */
/* @var $model Proposal */

$this->breadcrumbs=array(
    'Proposals'=>array('index'),
    'Manage',
);

$this->menu=array(
    //array('label'=>'List Proposal', 'url'=>array('index')),
    //array('label'=>'Create Proposal', 'url'=>array('create')),
);

Yii::app()->clientScript->registerScript('search', "
$('.search-button').click(function(){
    $('.search-form').toggle();
    return false;
});
$('.search-form form').submit(function(){
    $('#proposal-grid').yiiGridView('update', {
        data: $(this).serialize()
    });
});

```

```

        return false;
    });
    ");
    ?>

<h1>Daftar Proposal</h1>

<?php echo CHtml::link('Advanced
Search', '#', array('class'=>'search-button')); ?>
<div class="search-form" style="display:none">
<?php $this->renderPartial('_search', array(
    'model'=>$model,
)); ?>
</div><!-- search-form -->

<?php $this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'proposal-grid',
    'dataProvider'=>$model->search(),
    'filter'=>$model,
    'columns'=>array(
        'nim_mhs',
        'judul_proposal',
        array(
            'name'=>'file_proposal',
            'type'=>'raw',
            'value'=>'CHtml::link($data->file_proposal,
"http://" . $_SERVER["SERVER_NAME"] . Yii::app()->request->baseUrl
. "/images/foto/" . $data->file_proposal)',
        ),
        array(
            'class'=>'CButtonColumn',
        ),
    ),
)); ?>

```

create.php (Upload Proposal)

```

<?php
/* @var $this ProposalController */
/* @var $model Proposal */

$this->breadcrumbs=array(
    'Proposals'=>array('index'),
    'Create',
);

$this->menu=array(
    //array('label'=>'List Proposal', 'url'=>array('index')),
    //array('label'=>'Manage Proposal', 'url'=>array('admin')),
);
?>

<h1>Upload Proposal</h1>

<?php $this->renderPartial('_form', array('model'=>$model)); ?>

```

_form.php

```

<?php

```

```

/* @var $this ProposalController */
/* @var $model Proposal */
/* @var $form CActiveForm */
?>

<div class="form">

<?php $form=$this->beginWidget('CActiveForm', array(
    'id'=>'proposal-form',
    // Please note: When you enable ajax validation, make sure
the corresponding
    // controller action is handling ajax validation correctly.
    // There is a call to performAjaxValidation() commented in
generated controller code.
    // See class documentation of CActiveForm for details on
this.
    'enableAjaxValidation'=>false,
    'htmlOptions'=>array('enctype'=>'multipart/form-data'),
)); ?>

    <p class="note">Fields dengan tanda <span
class="required">*</span> harus diisi.</p>

    <?php echo $form->errorSummary($model); ?>

    <div class="row">
        <?php echo $form-
>hiddenField($model, 'nim_mhs', array('class'=>'User', 'value'=>Yii::
app()->user->name )); ?>
        <?php echo $form->error($model, 'nim_mhs'); ?>
    </div>

    <div class="row">
        <?php echo $form->labelEx($model, 'judul_proposal'); ?>
        <?php echo $form-
>textArea($model, 'judul_proposal', array('size'=>60, 'maxlength'=>25
0, 'row'=>4, 'cols'=>250)); ?>
        <?php echo $form->error($model, 'judul_proposal'); ?>
    </div>

    <div class="row">
        <?php echo $form->labelEx($model, 'file_proposal'); ?>
        <?php echo $form-
>fileField($model, 'file_proposal', array('size'=>30, 'maxlength'=>30
)); ?>
        <?php echo $form->error($model, 'file_proposal'); ?>
    </div>

    <div class="row buttons">
        <?php echo CHtml::submitButton($model->isNewRecord ?
'Upload' : 'Save'); ?>
    </div>

<?php $this->endWidget(); ?>
</div><!-- form -->

```

view.php

```
<?php
/* @var $this ProposalController */
/* @var $model Proposal */

$this->breadcrumbs=array(
    'Proposals'=>array('index'),
    $model->id_proposal,
);

$this->menu=array(
    //array('label'=>'List Proposal',
    'url'=>array('index'),'visible'=>Yii::app()->user-
>checkAccess('Kaprodi')),
    //array('label'=>'Upload Proposal', 'url'=>array('create')),
    array('label'=>'Hapus Proposal', 'url'=>'#',
    'linkOptions'=>array('submit'=>array('delete','id'=>$model-
>id_proposal),'confirm'=>'Anda yakin?'),'visible'=>Yii::app()-
>user->checkAccess('Kaprodi')),
    array('label'=>'Daftar Proposal',
    'url'=>array('admin'),'visible'=>Yii::app()->user-
>checkAccess('Kaprodi')),
);
?>

<h1>Detail Proposal <?php echo $model->nim_mhs; ?></h1>

<?php $this->widget('zii.widgets.CDetailView', array(
    'data'=>$model,
    'attributes'=>array(
        'nim_mhs',
        'judul_proposal',
        'file_proposal',
    ),
)); ?>
```

_view.php

```
<?php
/* @var $this ProposalController */
/* @var $data Proposal */
?>

<div class="view">

    <b><?php echo CHtml::encode($data-
>getAttributeLabel('nim_mhs')); ?></b>
    <?php echo CHtml::link(CHtml::encode($data->nim_mhs),
array('view', 'id'=>$data->id_proposal)); ?>
    <br />

    <b><?php echo CHtml::encode($data-
>getAttributeLabel('judul_proposal')); ?></b>
    <?php echo CHtml::encode($data->judul_proposal); ?>
    <br />
```

```

        <b><?php echo CHtml::encode($data->getAttributeLabel('file_proposal')); ?></b>
        <?php echo CHtml::link($data->file_proposal, "http://" .
        $_SERVER["SERVER_NAME"] . Yii::app()->request->baseUrl .
        "/images/foto/" . $data->file_proposal); ?>
        <br />

</div>

```

c. Controller

```

<?php

class ProposalController extends Controller
{
    /**
     * @var string the default layout for the views. Defaults to
     * '//layouts/column2', meaning
     * using two-column layout. See
     * 'protected/views/layouts/column2.php'.
     */
    public $layout='//layouts/column2';

    /**
     * @return array action filters
     */
    public function filters()
    {
        return array(
            // 'rights',
            'accessControl', // perform access control for
            CRUD operations
            'postOnly + delete', // we only allow deletion
            via POST request
        );
    }

    /**
     * Specifies the access control rules.
     * This method is used by the 'accessControl' filter.
     * @return array access control rules
     */
    public function accessRules()
    {
        return array(
            array('allow', // allow all users to perform
            'index' and 'view' actions

            'actions'=>array('admin','index','delete','view'),
            'hak akses'=>array('Kaprodi'),
            ),
            array('allow', // allow authenticated user to
            perform 'create' and 'update' actions
            'actions'=>array('create','view'),

```

```

        'hak akses'=>array('Mahasiswa'),
    ),
    array('allow', // allow admin user to perform
'admin' and 'delete' actions

        'actions'=>array('create','update','delete','admin','index',
'view'),

        'hak akses'=>array('SuperAdmin'),
    ),
    array('deny', // deny all users
        'users'=>array('*'),
    ),
);
}

/**
 * Displays a particular model.
 * @param integer $id the ID of the model to be displayed
 */
public function actionView($id)
{
    $this->render('view',array(
        'model'=>$this->loadModel($id),
    ));
}

/**
 * Creates a new model.
 * If creation is successful, the browser will be redirected
to the 'view' page.
 */
public function createAction()
{
    $model=new Proposal;

    // Uncomment the following line if AJAX validation is
needed
    // $this->performAjaxValidation($model);

    if(isset($_POST['Proposal']))
    {
        $model->attributes=$_POST['Proposal'];
        $model->file_proposal =
CUploadedFile::getInstance($model, 'file_proposal');

        if($model->save())
            if(strlen($model->file_proposal)>0)
                $model->file_proposal->saveAs(Yii::app()-
>basePath .'../images/foto/' . $model->file_proposal);
                $this->redirect(array('view','id'=>$model-
>id_proposal));
    }

    $this->render('create',array(
        'model'=>$model,
    ));
}

```

```

    }

    /**
     * Updates a particular model.
     * If update is successful, the browser will be redirected
to the 'view' page.
     * @param integer $id the ID of the model to be updated
     */
    public function actionUpdate($id)
    {
        $model=$this->loadModel($id);

        // Uncomment the following line if AJAX validation is
needed
        // $this->performAjaxValidation($model);

        if(isset($_POST['Proposal']))
        {
            $model->attributes=$_POST['Proposal'];
            $model->file_proposal =
CUUploadedFile::getInstance($model, 'file_proposal');

            if($model->save())
                if(strlen($model->file_proposal)>0)
                    $model->file_proposal->saveAs(Yii::app()-
>basePath .'../images/foto/' . $model->file_proposal);
                    $this->redirect(array('view','id'=>$model-
>id_proposal));
        }

        $this->render('update',array(
            'model'=>$model,
        ));
    }

    /**
     * Deletes a particular model.
     * If deletion is successful, the browser will be redirected
to the 'admin' page.
     * @param integer $id the ID of the model to be deleted
     */
    public function actionDelete($id)
    {
        $this->loadModel($id)->delete();

        // if AJAX request (triggered by deletion via admin
grid view), we should not redirect the browser
        if(!isset($_GET['ajax']))
            $this->redirect(isset($_POST['returnUrl']) ?
$_POST['returnUrl'] : array('admin'));
    }

    /**
     * Lists all models.
     */
    public function actionIndex()

```

```

    {
        $dataProvider=new CActiveDataProvider('Proposal');
        $this->render('index',array(
            'dataProvider'=>$dataProvider,
        ));
    }

    /**
     * Manages all models.
     */
    public function actionAdmin()
    {
        $model=new Proposal('search');
        $model->unsetAttributes(); // clear any default
values
        if(isset($_GET['Proposal']))
            $model->attributes=$_GET['Proposal'];

        $this->render('admin',array(
            'model'=>$model,
        ));
    }

    /**
     * Returns the data model based on the primary key given in
the GET variable.
     * If the data model is not found, an HTTP exception will be
raised.
     * @param integer $id the ID of the model to be loaded
     * @return Proposal the loaded model
     * @throws CHttpException
     */
    public function loadModel($id)
    {
        $model=Proposal::model()->findByPk($id);
        if($model===null)
            throw new CHttpException(404,'The requested page
does not exist.');
```

```

        return $model;
    }

    /**
     * Performs the AJAX validation.
     * @param Proposal $model the model to be validated
     */
    protected function performAjaxValidation($model)
    {
        if(isset($_POST['ajax']) &&
$_POST['ajax']=='proposal-form')
        {
            echo CActiveForm::validate($model);
            Yii::app()->end();
        }
    }
}

```


5. Dosen

a. Model

Dosen.php

```
<?php
```

```
/**
 * This is the model class for table "dosen".
 *
 * The followings are the available columns in table 'dosen':
 * @property integer $id_dosen
 * @property string $nama_dosen
 * @property string $nip
 * @property string $tgl_lahir
 * @property string $jabatan
 * @property string $gelar
 * @property string $pendidikan
 * @property string $bidang_keahlian
 * @property string $pangkat
 */
class Dosen extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'dosen';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those
        attributes that
        // will receive user inputs.
        return array(
            array('nama_dosen, nip, tgl_lahir, jabatan,
            gelar, pendidikan, bidang_keahlian, pangkat', 'required'),
            array('nama_dosen', 'length', 'max'=>50),
            array('nip, pangkat', 'length', 'max'=>30),
            array('jabatan', 'length', 'max'=>20),
            array('gelar', 'length', 'max'=>20),
            // The following rule is used by search().
            // @todo Please remove those attributes that
            should not be searched.
            array('id_dosen, nama_dosen, nip, tgl_lahir,
            jabatan, gelar, pendidikan, bidang_keahlian, pangkat', 'safe',
            'on'=>'search'),
        );
    }
}
```

```

/**
 * @return array relational rules.
 */
public function relations()
{
    // NOTE: you may need to adjust the relation name and
the related // class name for the relations automatically
generated below.
    return array(
    );
}

/**
 * @return array customized attribute labels (name=>label)
 */
public function attributeLabels()
{
    return array(
        'id_dosen' => 'Id Dosen',
        'nama_dosen' => 'Nama Dosen',
        'nip' => 'NIP',
        'tgl_lahir' => 'Tanggal Lahir',
        'jabatan' => 'Jabatan',
        'gelar' => 'Gelar',
        'pendidikan' => 'Pendidikan',
        'bidang_keahlian' => 'Bidang Keahlian',
        'pangkat' => 'Pangkat',
    );
}

/**
 * Retrieves a list of models based on the current
search/filter conditions.
 *
 * Typical usecase:
 * - Initialize the model fields with values from filter
form.
 * - Execute this method to get CActiveDataProvider instance
which will filter
 * models according to data in model fields.
 * - Pass data provider to CGridView, CListView or any
similar widget.
 *
 * @return CActiveDataProvider the data provider that can
return the models
 * based on the search/filter conditions.
 */
public function search()
{
    // @todo Please modify the following code to remove
attributes that should not be searched.

    $criteria=new CDbCriteria;

    $criteria->compare('id_dosen',$this->id_dosen);

```

```

        $criteria->compare('nama_dosen',$this->nama_dosen,true);
        $criteria->compare('nip',$this->nip,true);
        $criteria->compare('tgl_lahir',$this->tgl_lahir,true);
        $criteria->compare('jabatan',$this->jabatan,true);
        $criteria->compare('gelar',$this->gelar,true);
        $criteria->compare('pendidikan',$this->pendidikan,true);
        $criteria->compare('bidang_keahlian',$this->bidang_keahlian,true);
        $criteria->compare('pangkat',$this->pangkat,true);

        return new CActiveDataProvider($this, array(
            'criteria'=>$criteria,
        ));
    }

    /**
     * Returns the static model of the specified AR class.
     * Please note that you should have this exact method in all
     your CActiveRecord descendants!
     * @param string $className active record class name.
     * @return Dosen the static model class
     */
    public static function model($className=__CLASS__)
    {
        return parent::model($className);
    }
}

```

b. View

admin.php (Daftar Dosen)

```

<?php
/* @var $this DosenController */
/* @var $model Dosen */

$this->breadcrumbs=array(
    'Dosens'=>array('index'),
    'Manage',
);

$this->menu=array(
    //array('label'=>'List Dosen', 'url'=>array('index')),
    array('label'=>'Tambah Daftar',
        'url'=>array('create'),'visible'=>Yii::app()->user->checkAccess('Staff')),
);

Yii::app()->clientScript->registerScript('search', "
$('#search-button').click(function(){
    $('#search-form').toggle();
    return false;
});
$('#search-form form').submit(function(){
    $('#dosen-grid').yiiGridView('update', {
        data: $(this).serialize()
    })
});

```

```

        });
        return false;
    });
    ");
?>

<h1>Daftar Dosen</h1>

<?php $this->widget('application.extensions.tablesorter.Sorter',
array(
    'data'=>$records,
    'columns'=>array(
        'nama_dosen',
        'nip',
        'tgl_lahir',
        'bidang_keahlian',
        'jabatan',
        'gelar',
        /*
        'pendidikan',
        'pangkat',
        */
    ),
)); ?>

```

create.php

```

<?php
/* @var $this DosenController */
/* @var $model Dosen */

$this->breadcrumbs=array(
    'Dosens'=>array('index'),
    'Create',
);

$this->menu=array(
    //array('label'=>'List Dosen', 'url'=>array('index')),
    array('label'=>'Daftar Dosen', 'url'=>array('admin')),
);
?>

<h1>Tambah Data Dosen</h1>

<?php $this->renderPartial('_form', array('model'=>$model)); ?>

```

_form.php

```

<?php
/* @var $this DosenController */
/* @var $model Dosen */
/* @var $form CActiveForm */
?>

<div class="form">

<?php $form=$this->beginWidget('CActiveForm', array(

```

```

        'id'=>'dosen-form',
        // Please note: When you enable ajax validation, make sure
the corresponding
        // controller action is handling ajax validation correctly.
        // There is a call to performAjaxValidation() commented in
generated controller code.
        // See class documentation of CActiveForm for details on
this.
        'enableAjaxValidation'=>false,
    )); ?>

    <p class="note">Field dengan tanda <span
class="required">*</span> harus diisi.</p>
    <p class="note">Isi NIP Tanpa Spasi</p>

    <?php echo $form->errorSummary($model); ?>

    <div class="row">
        <?php echo $form->labelEx($model,'nama_dosen'); ?>
        <?php echo $form-
>textField($model,'nama_dosen',array('size'=>50,'maxlength'=>50));
?>
        <?php echo $form->error($model,'nama_dosen'); ?>
    </div>

    <div class="row">
        <?php echo $form->labelEx($model,'nip'); ?>
        <?php echo $form-
>textField($model,'nip',array('size'=>30,'maxlength'=>30)); ?>
        <?php echo $form->error($model,'nip'); ?>
    </div>

    <div class="row">
        <?php echo $form->labelEx($model,'tgl_lahir'); ?>
        <?php $this->widget('zii.widgets.jui.CJuiDatePicker',
            array(
                // you must specify name or
model/attribute
                'model'=>$model,
                'attribute'=>'tgl_lahir',
                'value' => $model->tgl_lahir,
                'options' => array(
                    'dateFormat'=>'yy-mm-dd',
                    'showOn'=>'button',
                    'yearRange'=>'-60',
                    'changeMonth' => 'true',
                    'changeYear' => 'true',
                    'constrainInput' => 'false',
                    'duration'=>'fast',
                    'showAnim' =>'slide',
                    //'showAnim' =>'fold',
                ),
                'htmlOptions'=>array('size'=>20,),
            )
        ); ?>
        <?php echo $form->error($model,'tgl_lahir'); ?>

```

```

        </div>

        <div class="row">
            <?php echo $form->labelEx($model,'jabatan'); ?>
            <?php echo $form-
>textField($model,'jabatan',array('size'=>20,'maxlength'=>20)); ?>
            <?php echo $form->error($model,'jabatan'); ?>
        </div>

        <div class="row">
            <?php echo $form->labelEx($model,'gelar'); ?>
            <?php echo $form-
>textField($model,'gelar',array('size'=>20,'maxlength'=>20)); ?>
            <?php echo $form->error($model,'gelar'); ?>
        </div>

        <div class="row">
            <?php echo $form->labelEx($model,'pendidikan'); ?>
            <?php echo $form-
>textField($model,'pendidikan',array('size'=>20,'maxlength'=>100))
; ?>
            <?php echo $form->error($model,'pendidikan'); ?>
        </div>

        <div class="row">
            <?php echo $form->labelEx($model,'bidang_keahlian');
?>
            <?php echo $form-
>textField($model,'bidang_keahlian',array('size'=>100,'maxlength'=>
>100)); ?>
            <?php echo $form->error($model,'bidang_keahlian'); ?>
        </div>

        <div class="row">
            <?php echo $form->labelEx($model,'pangkat'); ?>
            <?php echo $form-
>textField($model,'pangkat',array('size'=>30,'maxlength'=>30)); ?>
            <?php echo $form->error($model,'pangkat'); ?>
        </div>

        <div class="row buttons">
            <?php echo CHtml::submitButton($model->isNewRecord ?
'Tambah' : 'Save'); ?>
        </div>

<?php $this->endWidget(); ?>

</div><!-- form -->

```

view.php

```

<?php
/* @var $this DosenController */
/* @var $model Dosen */

$this->breadcrumbs=array(
    'Dosens'=>array('index'),

```

```

        $model->id_dosen,
    );

    $this->menu=array(
        //array('label'=>'List Dosen', 'url'=>array('index')),
        array('label'=>'Tambah Data',
            'url'=>array('create'),'visible'=>Yii::app()->user-
            >checkAccess('Staff')),
        array('label'=>'Ubah Data', 'url'=>array('update',
            'id'=>$model->id_dosen),'visible'=>Yii::app()->user-
            >checkAccess('Staff')),
        array('label'=>'Hapus Data', 'url'=>'#',
            'linkOptions'=>array('submit'=>array('delete','id'=>$model-
            >id_dosen),'confirm'=>'Anda yakin?'),'visible'=>Yii::app()->user-
            >checkAccess('Staff')),
        array('label'=>'Daftar Dosen', 'url'=>array('admin')),
    );
?>

<h1>Detail Dosen <?php echo $model->nip; ?></h1>

<?php $this->widget('zii.widgets.CDetailView', array(
    'data'=>$model,
    'attributes'=>array(
        'nama_dosen',
        'nip',
        'tgl_lahir',
        'jabatan',
        'gelar',
        'pendidikan',
        'bidang_keahlian',
        'pangkat',
    ),
)); ?>

```

_view.php

```

<?php
/* @var $this DosenController */
/* @var $data Dosen */
?>

<div class="view">

    <b><?php echo CHtml::encode($data-
    >getAttributeLabel('nip')); ?></b>
    <?php echo CHtml::link(CHtml::encode($data->nip),
    array('view', 'id'=>$data->id_dosen)); ?>
    <br />

    <b><?php echo CHtml::encode($data-
    >getAttributeLabel('nama_dosen')); ?></b>
    <?php echo CHtml::encode($data->nama_dosen); ?>
    <br />

    <b><?php echo CHtml::encode($data-
    >getAttributeLabel('tgl_lahir')); ?></b>

```

```

        <?php echo CHtml::encode($data->tgl_lahir); ?>
        <br />

        <b><?php echo CHtml::encode($data-
>getAttributeLabel('bidang_keahlian')); ?></b>
        <?php echo CHtml::encode($data->bidang_keahlian); ?>
        <br />

        <b><?php echo CHtml::encode($data-
>getAttributeLabel('jabatan')); ?></b>
        <?php echo CHtml::encode($data->jabatan); ?>
        <br />

        <b><?php echo CHtml::encode($data-
>getAttributeLabel('gelar')); ?></b>
        <?php echo CHtml::encode($data->gelar); ?>
        <br />

        <?php /*
        <b><?php echo CHtml::encode($data-
>getAttributeLabel('pendidikan')); ?></b>
        <?php echo CHtml::encode($data->pendidikan); ?>
        <br />

        <b><?php echo CHtml::encode($data-
>getAttributeLabel('pangkat')); ?></b>
        <?php echo CHtml::encode($data->pangkat); ?>
        <br />

        */ ?>

</div>

```

c. Controller

DosenController.php

```

<?php

class DosenController extends Controller
{
    /**
     * @var string the default layout for the views. Defaults to
     * '//layouts/column2', meaning
     * using two-column layout. See
     * 'protected/views/layouts/column2.php'.
     */
    public $layout='//layouts/column2';

    /**
     * @return array action filters
     */
    public function filters()
    {
        return array(
            // 'rights',
            'accessControl', // perform access control for
            CRUD operations

```



```

        'postOnly + delete', // we only allow deletion
via POST request
    );
}

/**
 * Specifies the access control rules.
 * This method is used by the 'accessControl' filter.
 * @return array access control rules
 */
public function accessRules()
{
    return array(
        array('allow', // allow authenticated user to
perform 'create' and 'update' actions
            'actions'=>array('index','view','admin'),
            'users'=>array('@'),
        ),
        array('allow', // allow admin user to perform
'admin' and 'delete' actions
            'actions'=>array('create','update','delete'),
            'hak
aksess'=>array('Staff','Kaprodi','SuperAdmin'),
        ),
        array('deny', // deny all users
            'users'=>array('*'),
        ),
    );
}

/**
 * Displays a particular model.
 * @param integer $id the ID of the model to be displayed
 */
public function actionView($id)
{
    $this->render('view',array(
        'model'=>$this->loadModel($id),
    ));
}

/**
 * Creates a new model.
 * If creation is successful, the browser will be redirected
to the 'view' page.
 */
public function createAction()
{
    $model=new Dosen;

    // Uncomment the following line if AJAX validation is
needed
    // $this->performAjaxValidation($model);

    if(isset($_POST['Dosen']))

```

```

        {
            $model->attributes=$_POST['Dosen'];
            if($model->save())
                $this->redirect(array('view','id'=>$model-
>id_dosen));
        }

        $this->render('create',array(
            'model'=>$model,
        ));
    }

    /**
     * Updates a particular model.
     * If update is successful, the browser will be redirected
to the 'view' page.
     * @param integer $id the ID of the model to be updated
     */
    public function actionUpdate($id)
    {
        $model=$this->loadModel($id);

        // Uncomment the following line if AJAX validation is
needed
        // $this->performAjaxValidation($model);

        if(isset($_POST['Dosen']))
        {
            $model->attributes=$_POST['Dosen'];
            if($model->save())
                $this->redirect(array('view','id'=>$model-
>id_dosen));
        }

        $this->render('update',array(
            'model'=>$model,
        ));
    }

    /**
     * Deletes a particular model.
     * If deletion is successful, the browser will be redirected
to the 'admin' page.
     * @param integer $id the ID of the model to be deleted
     */
    public function actionDelete($id)
    {
        $this->loadModel($id)->delete();

        // if AJAX request (triggered by deletion via admin
grid view), we should not redirect the browser
        if(!isset($_GET['ajax']))
            $this->redirect(isset($_POST['returnUrl']) ?
$_POST['returnUrl'] : array('admin'));
    }

```

```

/**
 * Lists all models.
 */
public function actionIndex()
{
    $dataProvider=new CActiveDataProvider('Dosen');
    $this->render('index',array(
        'dataProvider'=>$dataProvider,
    ));
}

/**
 * Manages all models.
 */
public function actionAdmin()
{
    $records=Dosen::model()->findAll();
    $this->render('admin',array(
        'records'=>$records,
    ));
}

/**
 * Returns the data model based on the primary key given in
the GET variable.
 * If the data model is not found, an HTTP exception will be
raised.
 * @param integer $id the ID of the model to be loaded
 * @return Dosen the loaded model
 * @throws CHttpException
 */
public function loadModel($id)
{
    $model=Dosen::model()->findByPk($id);
    if($model===null)
        throw new CHttpException(404,'The requested page
does not exist.');
```

return \$model;

```

}

/**
 * Performs the AJAX validation.
 * @param Dosen $model the model to be validated
 */
protected function performAjaxValidation($model)
{
    if(isset($_POST['ajax']) && $_POST['ajax']=='dosen-
form')
    {
        echo CActiveForm::validate($model);
        Yii::app()->end();
    }
}
}

```

6. Hak akses

a. Model

AuthAssignment.php

```
<?php
```

```
/**
 * This is the model class for table "AuthAssignment".
 *
 * The followings are the available columns in table
 'AuthAssignment':
 * @property string $itemname
 * @property string $userid
 * @property string $bizrule
 * @property string $data
 *
 * The followings are the available model relations:
 */
class AuthAssignment extends CActiveRecord
{
    /**
     * Returns the static model of the specified AR class.
     * @return AuthAssignment the static model class
     */
    public static function model($className=__CLASS__)
    {
        return parent::model($className);
    }

    /**
     *
     * @desc check if User is assigned to an AuthItem
     * @param string $username from table user.username
     * @param string $itemname from table
authAssignment.itemname
     * @return boolean
     */
    public static function userIsAssigned($userid, $itemname)
    {
        return (AuthAssignment::model()-
>findByAttributes(array('userid'=>$userid, 'itemname'=>$itemname))
!= null ? true : false);
    }

    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'AuthAssignment';
    }

    /**
     * @return array validation rules for model attributes.

```

```

        */
        public function rules()
        {
            // NOTE: you should only define rules for those
attributes that
            // will receive user inputs.
            return array(
                array('itemname, userid', 'required'),
                array('itemname, userid', 'length', 'max'=>64),
                array('bizrule, data', 'safe'),
                // The following rule is used by search().
                // Please remove those attributes that should
not be searched.
                //array('itemname, userid, bizrule, data',
'safe', 'on'=>'search'),
            );
        }

        /**
         * @return array relational rules.
         */
        public function relations()
        {
            return array(
                'user'=>array(
                    self::HAS_MANY,
                    'User',
                    'username'
                )
            );
        }

        /**
         * @return array customized attribute labels (name=>label)
         */
        public function attributeLabels()
        {
            return array(
                'itemname' => 'Itemname',
                'userid' => 'Username',
                'bizrule' => 'BizRule',
                'data' => 'Data',
            );
        }
    }
}

```

AuthItem.php

```

<?php

/**
 * This is the model class for table "AuthItem".
 *
 * The followings are the available columns in table 'AuthItem':
 * @property string $name

```

```

* @property integer $type
* @property string $description
* @property string $bizrule
* @property string $data
*
* The followings are the available model relations:
*/
class AuthItem extends CActiveRecord
{
    /**
     * Returns the static model of the specified AR class.
     * @return AuthItem the static model class
     */
    public static function model($className=__CLASS__)
    {
        return parent::model($className);
    }

    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'AuthItem';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        if(isset($_GET['item'])) return array(array('name',
'safe'));
        return array(
            array('name, type', 'required'),
            array('type', 'numerical', 'integerOnly'=>true),
            array('name', 'length', 'max'=>64),
            array('description, bizrule, data', 'safe'),
            // The following rule is used by search().
            // Please remove those attributes that should
not be searched.
            array('name, type, description, bizrule, data',
'safe', 'on'=>'search'),
        );
    }

    /**
     * @return array relational rules.
     */
    public function relations()
    {
        return array(
            /*
             * These both are used to find AuthItem they are
not bound in AuthItemChild
             * and don't appear in the RBAC Tree.

```

```

        * Relation 'childs' is exactly the same as
Relation 'unbound Child'
        * look in AuthItemChild::_findUnboundItems()
        */
        'unboundParents' => array( // LEFT OUTER JOIN
AuthItemChild AS unboundParents ON t.name = unboundParents.parent
            self::HAS_MANY,
            'AuthItemChild',
            'parent',
            'select' => 'unboundParents.parent
AS uParent',
        ),
        'unboundChilds' => array( // LEFT OUTER JOIN
AuthItemChild AS unboundChilds ON t.name = unboundChilds.child
            self::HAS_MANY,
            'AuthItemChild',
            'child',
            'select' => 'unboundChilds.child AS
uChild',
        ),
    );
}

/**
 * @desc If Items are not listed in table AuthItemChild,
they don't appear in the RBAC Tree.
 * @desc So we have to find them seperately
 *
 */
public function findUnboundItems()
{
    /*
    * SELECT `t`.`name` AS `t0_c0`,
    *         `t`.`type` AS `t0_c1`,
    *         `t`.`description` AS `t0_c2`,
    *         `t`.`bizrule` AS `t0_c3`,
    *         `t`.`data` AS `t0_c4`,
    *         `unboundParents`.`parent` AS `t1_c0`,
    *         `unboundParents`.`child` AS `t1_c1`,
    *         `unboundChilds`.`parent` AS `t2_c0`,
    *         `unboundChilds`.`child` AS `t2_c1`
    * FROM `AuthItem` `t`
    * LEFT OUTER JOIN `AuthItemChild` `unboundParents`
    *         ON (`unboundParents`.`parent`=`t`.`name`)
    * LEFT OUTER JOIN `AuthItemChild` `unboundChilds`
    *         ON (`unboundChilds`.`child`=`t`.`name`)
    * WHERE (unboundParents.parent is Null AND
unboundChilds.child is NULL)
    * ORDER BY t.parent
    */
    $result = $this->findAll(array(
        'with' => array('unboundParents',
'unboundChilds'),
        'condition' => 'unboundParents.parent is
Null AND unboundChilds.child is NULL',
        'order' => 't.type ASC, t.name DESC'
    ));
}

```

```

        )
    );
    return $result;
}

/**
 * @return array customized attribute labels (name=>label)
 */
public function attributeLabels()
{
    return array(
        'name' => 'Item Name',
        'type' => 'Type',
        'description' => 'Description',
        'bizrule' => 'Bizrule',
        'data' => 'Data',
    );
}

/**
 * Retrieves a list of models based on the current
search/filter conditions.
 * @return CActiveDataProvider the data provider that can
return the models based on the search/filter conditions.
 */
public function search()
{
    // Warning: Please modify the following code to remove
attributes that
    // should not be searched.

    $criteria=new CDbCriteria;

    $criteria->compare('name',$this->name,true);
    $criteria->compare('type',$this->type);
    $criteria->compare('description',$this-
>description,true);
    $criteria->compare('bizrule',$this->bizrule,true);
    $criteria->compare('data',$this->data,true);

    return new CActiveDataProvider(get_class($this),
array(
        'criteria'=>$criteria,
    ));
}
}

```

AuthItemChild.php

<?php

```

/**
 * This Class manage the Access Items Tree
 * It does create, move and delete AuthItems
 *
 * The main action is to build an array Tree with AuthItems
 */

```



```

*/
class AuthItemChild extends CActiveRecord
{

    public $parent;
    public $child;

    /**
     * Returns the static model of the specified AR class.
     * @return AuthItemChild the static model class
     */
    public static function model($className=__CLASS__)
    {
        return parent::model($className);
    }

    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'AuthItemChild';
    }

    /**
     * @see framework/base/CModel::rules()
     */
    public function rules()
    {
        return array(
            array('parent, child', 'required'),
            // array('parent, child', 'safe'), // is a child
of all other validations
        );
    }

    /**
     * @return array relational rules.
     */
    public function relations()
    {
        return array(
            /*
             * These both are used to build the RBAC Tree
             * look in $this->getItemTree()
             */
            'parents' => array( // LEFT OUTER JOIN
AuthItemChild AS parents ON AuthItemChild.parent = parents.child
                self::HAS_MANY,
                'AuthItemChild',
                'child',
            ),
            'childs' => array( // LEFT OUTER JOIN AuthItem
AS childs ON (AuthItemChild.child = childs.name)
                self::BELONGS_TO,

```

```

        'AuthItem',
        'child',
    ),
);
}

}

```

User.php

```

<?php

/**
 *
 * feel free to mod this Class
 *
 */
class User extends CActiveRecord
{

    public static function model($className=__CLASS__)
    {
        return parent::model($className);
    }

    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        // DO NOT CHANGE
        return Yii::app()->controller->module->tableUser;

        // or use your known table name like
        // return 'TableUser';
    }

    public function rules()
    {
        return array();
    }

    public function relations()
    {
        return array();
    }

    public function attributeLabels()
    {
        return array();
    }
}

```

b. View

1) Assignments

index.php

```
<?php
$this->breadcrumbs=array(
    'Assignments'=>$this->createUrl('//rbac/assignment'),
);
?>

<h1><b>Hak akses Pengguna</b> - Userlist</h1><br>
<?php
// form errors and success
$model->renderMessages();
// pager
$this->widget('CLinkPager', array(
    'pages' => $pages,
))
?><br><br><?php
// Form Userlist with Hak akses
$this->renderPartial('_listForm', array('getVars'=>$getVars,
'users'=>$users, 'colUsername'=>$colUsername,
'colUserId'=>$colUserId));

// pager
$this->widget('CLinkPager', array(
    'pages' => $pages,
))
?>
```

manage.php

```
<?php
$this->breadcrumbs=array(
    'Assignments'=>$this->createUrl('//rbac/assignment'),
    'Manage',
);
?>

<h1>Hak akses Assignments - Assign/Eject Hak akses</h1><br>
<?php
// form errors and success
$model->renderMessages();
?>

<b>Assign/Eject Hak akses from:
<table>
    <tr>
        <th>User</th>
        <th>Email</th>
    </tr>
    <tr>
        <td><?php echo CHtml::encode($manageUser-
>$colUsername)?></td>
        <td><?php echo CHtml::encode($manageUser-
>$colEmail)?></td>
```

```

        </tr>
    </table>
    <b>Or back to: <a href="<?php echo $this->createUrl('//rbac/assignment', $getVars)?>">Userlist</a>,
    <a href="<?php echo $this->createUrl('//rbac/assignment/edit',
    array_merge(array('userid'=>$manageUser->$colUserid), $getVars))
    ?>">edit Assignments</a></b>.
</b>
<?php

if($manageUser && $displayHelper->hasItems()):
    $this->renderPartial('_treeForm',
        array(
            'displayHelper'=>$displayHelper,
            'getVars'=>$getVars,
            'manageUser'=>$manageUser,
            'colUsername'=>$colUsername,
            'colUserid'=>$colUserid
        )
    );
endif;
?>
<br><b>Note:<br></b>
After a Buisness Rule the AccessController ends with the Buisness
Rule Result and does not climp up to Parent any more.<br>
Check Access steps in Detail: <br>
1.) if User has requested Item Assigned do 2) - else do 4)<br>
2.) if requested User Assignment Item has Buisness Rule check and
end - else do 3)<br>
3.) if requested RBAC Item has Buisness Rule check and end - else
end with true<br>
4.) if requested Item has parent do 1) with parent as requested
Item - else end with false<br><br>
The Data field is not used while checking access. Feel free to use
it fore your needs.

```

_listForm.php

```

<div>
<?php echo CHtml::beginForm($this->createUrl('//rbac/assignment'),
'get') ?>
    <table style="width:70%">
        <tr>
            <th><?php echo CHtml::textField('s1',
            $getVars['s1'], array('size'=>'10',
            'onChange'=>'this.form.submit()')) ?></th>
            <th><?php echo CHtml::textField('s2',
            $getVars['s2'], array('size'=>'10',
            'onChange'=>'this.form.submit()')) ?></th>
            <th><?php echo CHtml::textField('s3',
            $getVars['s3'], array('size'=>'15',
            'onChange'=>'this.form.submit()')) ?></th>
            <th><?php echo CHtml::textField('s4',
            $getVars['s4'], array('size'=>'10',
            'onChange'=>'this.form.submit()')) ?></th>
        </tr>
    </table>
<?php

```

```

$lastUser='';
$style='style="border-top:1px solid #cdcdcd"';
foreach($users as $user):
    if($user[$colUserId] == $lastUser) $style='';
?>
        <tr>
            <td <?php echo $style ?> ><?php echo ($lastUser !=
$user[$colUserId] ? '<a href="'. $this-
>createUrl('//rbac/assignment/manage',
array_merge(array('userid'=>$user[$colUserId]),$getVars)).'">'.CHt
ml::encode($user[$colUsername]).'</a>' : '&nbsp;')?></td>
            <td <?php echo $style ?> ><?php echo '<a
href="'. $this->createUrl('//rbac/assignment/edit',
array_merge(array('userid'=>$user[$colUserId]),$getVars)).'">'.CHt
ml::encode($user['itemname']).'</a>' ?></td>
            <td <?php echo $style ?> nowrap ><?php echo
nl2br(CHtml::encode($user['bizrule'])) ?></td>
            <td <?php echo $style ?> ><?php echo
CHtml::encode($user['data']) ?></td>
        </tr>
<?php
    $lastUser=$user[$colUserId];
    $style='style="border-top:1px solid #cdcdcd"';
endforeach;
?>
</table>
<?php echo CHtml::endForm() ?>
</div>

```

edit.php

```

<?php
$this->breadcrumbs=array(
    'Assignments'=>$this->createUrl('//rbac/assignment'),
    'Edit',
);
?>
<h1>Edit Hak akses Assignments</h1><br>
<?php
// form errors and success
$model->renderMessages();
?>
<h3>from:</h3>
<table>
    <tr>
        <th>Username</th>
        <th>Email</th>
    </tr>
    <tr>
        <td><?php echo CHtml::encode($user->$colUsername) ?>
        <td><?php echo CHtml::encode($user->$colEmail) ?>
    </td>
    </tr>
</table>
<b>Or back to: <a href="<?php echo $this-
>createUrl('//rbac/assignment', $getVars)?>">Userlist</a>,

```

```

<a href="php echo $this-&gt;createUrl('//rbac/assignment/manage',
array_merge(array('userid'=&gt;$user-
&gt;$colUserId),$getVars))?">Assign/Eject Hak akses</a></b><br><br>
<?php
if(count($assignments)):
    $this->renderPartial('_editForm', array('getVars'=>$getVars,
'assignments'=>$assignments, 'user'=>$user,
'colUsername'=>$colUsername, 'colUserId'=>$colUserId));
else:
?>User has no <a href="php echo $this-
&gt;createUrl('//rbac/assignment/manage',
array_merge(array('userid'=&gt;$user-
&gt;$colUserId),$getVars))?">Assignments</a></b><br><br>
<?php
endif;

```

_editForm.php

```

<div>
<?php echo CHtml::beginForm($this-
>createUrl('//rbac/assignment/edit', $getVars), 'post') ?>
<?php echo CHtml::hiddenField('username', $user->$colUsername) ?>
<?php echo CHtml::hiddenField('userid', $user->$colUserId) ?>
    <table>
        <tr>
            <th>Hak akses</th>
            <th>Buisness Rule</th>
            <th>Data</th>
        </tr>
        <?php foreach($assignments as $assignment): ?>
            <tr>
                <td><b><?php echo CHtml::encode($assignment-
>itemname) ?></b></td>
                <td><?php echo
CHtml::textArea("assignments[\".$assignment->itemname.\"][bizrule]",
$assignment->bizrule, array('cols'=>60, 'rows'=>5))?> </td>
                <td><?php echo
CHtml::textArea("assignments[\".$assignment->itemname.\"][data]",
$assignment->data, array('cols'=>30, 'rows'=>5))?> </td>
            </tr>
        <?php endforeach ?>
    </table>
    <span style="padding-left:4em"><?php echo
CHtml::submitButton('Save Changes') ?></span>
    <?php echo CHtml::endForm() ?>
</div>

```

2) RBAC Tree

index.php

```

<?php
$this->breadcrumbs=array(
    'Manage Rbac Tree'=>$this->createUrl('//rbac/rbac'),
);
?>
<h1>Daftar Hak akses</h1>

```

```

<?php

// form errors and success
$model->renderMessages();

// edit form
if($editItem !== null || !$displayHelper->hasItems()):
    if(!$displayHelper->hasItems()):
        $editItem=new AuthItem;
    endif;
    $this->renderPartial('_editForm',
array('editItem'=>$editItem, 'displayHelper'=>$displayHelper,
'nameFilter'=>$nameFilter));
endif;

// eject form
if(count($ejectItemsGet)):
    $this->renderPartial('_ejectForm',
array('ejectItemsGet'=>$ejectItemsGet,
'displayHelper'=>$displayHelper));
endif;

// edit form
if($treeFormat == 'renderItemTree'):
    ?>
    <br>
    <a href="<?php echo $this-
>createUrl('//rbac/rbac',array('createNew'=>1)) ?>">Tambah Hak
akses Baru</a>
    <br>
    <?php
endif;
if($displayHelper->hasItems()):
    $this->renderPartial('_treeForm',
array('displayHelper'=>$displayHelper,
'treeFormat'=>$treeFormat));
endif;
?>
<br><b>Note:<br></b>
After a Buisness Rule the AccessController ends with the Buisness
Rule Result and does not climp up to Parent any more.<br>
Check Access steps in Detail: <br>
1.) if User has requested Item Assigned do 2) - else do 4)<br>
2.) if requested User Assignment Item has Buisness Rule check and
end - else do 3)<br>
3.) if requested RBAC Item has Buisness Rule check and end - else
end with true<br>
4.) if requested Item has parent do 1) with parent as requested
Item - else end with false<br><br>
The Data field is not used while checking access. Feel free to use
it fore your needs.

```

_treeForm.php

```

<?php

```

```

if($treeFormat == 'renderItemTree'):
?>
    <div>
        <?php echo CHtml::beginForm($this->createUrl('//rbac/rbac/move'), 'post') ?>
        <br>
        <?php echo CHtml::submitButton('Move Item') ?>
        <div style="width: 90%">
            <?php echo $displayHelper->renderTree(); ?>
        </div>
        <br>
        <?php echo CHtml::submitButton('Move Item') ?>
    <?php echo CHtml::endForm() ?>
</div>
<br>
<a href="<?php echo $this->createUrl('//rbac/rbac',array('generateCode'=>1)) ?>">Generate
PHP Code</a>
<br>
<?php
elseif($treeFormat == 'renderCode'):
?>
    <br>
    <?php
        echo $displayHelper->renderTree("\$auth=Yii::app()-
>authManager;\n");
    ?>
    <br>
    <a href="<?php echo $this->createUrl('//rbac/rbac')
?>">Return to Tree View</a>
    <br>
<?php
elseif($treeFormat == 'renderItemInfo'):
?>
    <br>
    <?php
        echo $displayHelper->renderTree();
    ?>
<?php
endif;

```

_editForm.php

```

<div>

    <script type="text/javascript">
        confirmDelete = function(_t){
            if(confirm("Really delete Item?")){
                _t.form.submit();
            }
        }
    </script>

    <?php echo CHtml::errorSummary($editItem, '<div
id="rbacError">', '</div>'); ?>
    <?php echo CHtml::beginForm($this->createUrl('rbac/edit'),
'post') ?>

```



```

        <?php echo CHtml::hiddenField('oldName', $editItem->name);
?>
        <table style="width: 50%">
            <tr>
                <td>Item Name:</td>
                <td>
                    <?php echo CHtml::textField
('editItem[name]', $editItem->name); ?>
                </td>
                <td><span style="color:#cc0000">Required (min.2
Chars)</span><br>Filter: <?php echo $nameFilter ?></td>
            </tr>
            <tr>
                <td>Type:</td>
                <td colspan="2">
                    <?php echo CHtml::dropDownList
('editItem[type]', $editItem->type,
array('Operation', 'Task', 'Hak akses')) ?><br/>
                </td>
            </tr>
            <tr>
                <td>Description:</td>
                <td colspan="2">
                    <?php echo CHtml::textArea
('editItem[description]', $editItem->description,
array('cols' => 70, 'rows' => 4)) ?><br/>
                </td>
            </tr>
            <tr>
                <td>Bizrule:</td>
                <td colspan="2">
                    <?php echo CHtml::textArea
('editItem[bizrule]', $editItem->bizrule,
array('cols' => 70, 'rows' => 4)) ?><br/>
                </td>
            </tr>
            <tr>
                <td>Data:</td>
                <td colspan="2">
                    <?php echo CHtml::textArea
('editItem[data]', $editItem->data,
array('cols' => 70, 'rows' => 4)); ?><br/>
                </td>
            </tr>
            <tr>
                <td colspan="3">
                    <?php
                    if(!$displayHelper->hasItems() ||
isset($_GET['createNew'])):
                        echo CHtml::submitButton('Tambah Hak
akses', array('name'=>'createItem'));
                    else:
                        ?>Edit/Create/Delete:<br/>
                        <table>
                            <tr>
                                <td>

```

```

                                <?php echo
CHtml::submitButton('Update Hak akses',
    array('name'=>'updateItem')) ?>
                                </td>
                                <td>
                                <?php echo
CHtml::submitButton('Simpan Sebagai Hak akses Baru',
    array('name'=>'createItem')) ?>
                                </td>
                                <td>
                                <?php echo
CHtml::submitButton('Hapus Hak akses '.$editItem->name.'',
    array('name'=>'deleteItem',
    'onClick'=>"confirmDelete(this)")) ?>
                                </td>
                                </tr>
                                </table>
                                <b>or <a href="<?php echo $this-
>createUrl('//rbac/rbac') ?>">close</a> input Fields.</b>
                                <?php endif ?>
                                </td>
                                </tr>
                                </table>
                                <?php echo CHtml::endForm(); ?>
</div>

```

_ejectForm.php

```

<?php ?>
<div style="padding: 2em; clear: both; border-left: 2px solid
#cccccc; border-top: 2px solid #cccccc; border-bottom: 2px solid
#cccccc" >
    <?php echo CHtml::beginForm($this->createUrl('rbac/eject'),
    'post') ?>
        <?php echo CHtml::hiddenField ('eject[parent]',
        $ejectItemsGet['parent']->name); ?>
        <?php echo CHtml::hiddenField ('eject[child]',
        $ejectItemsGet['child']->name); ?>
        <h3>Eject Child <i><?php echo $ejectItemsGet['child']->
name ?></i> from Parent <i><?php echo $ejectItemsGet['parent']->
name ?></i></h3>
        <?php
            echo $displayHelper->treeFormat['groupIn'];
            echo $displayHelper-
>renderItemEject($ejectItemsGet['parent']);
            echo $displayHelper->treeFormat['groupIn'];
            echo $displayHelper-
>renderItemEject($ejectItemsGet['child']);
        ?>
        <br />
        <?php echo CHtml::submitButton ('Eject Item'); ?>
        <?php
            echo $displayHelper->treeFormat['groupOut'];
            echo $displayHelper->treeFormat['groupOut'];
        ?>
    <?php echo CHtml::endForm() ?>

```

</div>

c. Controller

DefaultController.php

```
<?php
class DefaultController extends RBACBaseController
{
    public function actionIndex()
    {
        $this->redirect(array('//rbac/rbac'));
    }
}
```

AssignmentController.php

```
<?php

class AssignmentController extends RBACBaseController
{

    // user List
    public $users;
    public $authAssignments; // yii crashes by name $assignments
in CWebApplication in Line 391: if(!classExists(...)

    // manage user Assignments
    public $manageUser;

    // edit Assignments from User
    public $editUser;
    public $assignments;

    // CPagination
    public $paginate;
    public $usersPerPage=20;

    // search
    public $searchFields;
    public $defaultSearchFields=array(
        's1'=>'Username', // displayed default value
        's2'=>'Hak akses',
        's3'=>'Buisness Rule',
        's4'=>'Data'
    );

    /**
     * Lists all models.
     */
    public function actionIndex()
    {
        $this->checkAccess('RbacAssignmentViewer', true);
```

```

        $this->doRender('index', array(
            'assignUser'=>$this->manageUser,
            'users' => $this->_getUserlist(),
            'pages' => $this->paginate,
            'getVars' => $this->getGetVars()
        ));
        Yii::app()->end();
    }

    /**
     *
     * @desc
     */
    private function _getUserlist()
    {
        /**
         * get changeable table- and columnnames
         */
        $module=Yii::app()->controller->module;
        $tblUser=$module->tableUser;
        $colUsername=$module->columnUsername;
        $columnUserid=$module->columnUserid;

        /*
         * this is a very ugly yii workaround
         * of Paginate Tables with HAS_MANY Joins and
Conditions
         *
         * I do this with two selects:
         * one for count, limit and offset
         * and one for the Item select
         * In both we have to use the conditions
         */

        // create criteria for count, limit and offset select
        $criteria=new CDbCriteria(
            array(
                'with'=>array('assignment'),
                'order'=>'t.$colUsername ASC'
            )
        );

        // get Getvars and diff $this->searchFields and $this-
>defaultSearchFields
        $this->_getSearchFields();

        // combine getvars with tablenames
        $columns=array(
            's1'=>'t'.$colUsername,
            's2'=>'assignment.itemname',
            's3'=>'assignment.bizrule',
            's4'=>'assignment.data');

        // array for second itemselect

```

```

        $conditions=array();

        // get the conditions from getvars s1, s2, s3 and s4
        if(isset($this->searchFields['s1']) && $this->searchFields['s1'] != $this->defaultSearchFields['s1'])
        {
            $criteria->addSearchCondition($columns['s1'],
            $this->searchFields['s1'], true, 'AND', 'LIKE');
            $conditions['s1']="$tblUser.$colUsername LIKE
:s1";
        }
        if(isset($this->searchFields['s2']) && $this->searchFields['s2'] != $this->defaultSearchFields['s2'])
        {
            $criteria->addSearchCondition($columns['s2'],
            $this->searchFields['s2'], true, 'AND', 'LIKE');
            $conditions['s2']="AuthAssignment.itemname LIKE
:s2";
        }
        if(isset($this->searchFields['s3']) && $this->searchFields['s3'] != $this->defaultSearchFields['s3'])
        {
            $criteria->addSearchCondition($columns['s3'],
            $this->searchFields['s3'], true, 'AND', 'LIKE');
            $conditions['s3']="AuthAssignment.bizrule LIKE
:s3";
        }
        if(isset($this->searchFields['s4']) && $this->searchFields['s4'] != $this->defaultSearchFields['s4'])
        {
            $criteria->addSearchCondition($columns['s4'],
            $this->searchFields['s4'], true, 'AND', 'LIKE');
            $conditions['s4']="AuthAssignment.data LIKE
:s4";
        }

        // get the DB for the count select
        $db=Yii::app()->getDb();

        // the first half query
        $sql="SELECT COUNT(*)
        FROM $tblUser
        LEFT JOIN AuthAssignment
        ON AuthAssignment.userid =
        $tblUser.$colUsername
        ";

        // add the WHERE condition
        if(count($conditions)){
            $sql.=" WHERE (" .implode(' AND ',
$conditions).") ";
        }

        // create the command
        $command=$db->createCommand($sql);

```

```

        // bind the values
        foreach($conditions as $key => $value)
        {
            $command->bindValue($key, '%'.$this->searchFields[$key].'%');
        }
        // and finally get the REAL count of Items
        $count=$command->queryScalar();

        // create the pagination
        $this->paginate=new CPagination($count);
        $this->paginate->pageSize=$this->usersPerPage;

        // after this we will find limit and offset in the
criteria
        $this->paginate->applyLimit($criteria);

        // now begin the second select for the Items
        $sql=" SELECT $tblUser.$columnUserid,
        $tblUser.$colUsername, AuthAssignment.itemname,
        AuthAssignment.bizrule, AuthAssignment.data
            FROM $tblUser
            LEFT JOIN AuthAssignment
            ON AuthAssignment.userid =
        $tblUser.$columnUserid
            ";

        // add the WHERE condition
        if(count($conditions)){
            $sql.=" WHERE (".$implode(' AND ',
$conditions).") ";
        }
        // add ORDER
        $sql.=" ORDER BY $tblUser.$colUsername ASC,
        AuthAssignment.itemname";

        // add limit and offset from $criteria
        $sql.=" LIMIT ".$criteria->offset.','.$criteria->limit;

        // create the command
        $command=$db->createCommand($sql);

        // bind the values
        foreach($conditions as $key => $value)
        {
            $command->bindValue($key, '%'.$this->searchFields[$key].'%');
        }

        // and here the are
        $this->users=$command->queryAll();
        return $this->users;
    }

```

```

/**
 *
 * @desc
 */
private function _getSearchFields()
{
    if($this->searchFields !== null) return;
    if(isset($_GET['sl']))
    {
        $resource = $_GET;
    }elseif(isset($_POST['search'])) {
        $resource = $_POST['search'];
    }else{
        $this->searchFields=$this->defaultSearchFields;
        return;
    }
    foreach($this->defaultSearchFields as $k => $v)
    {
        if(!isset($resource[$k]))
        {
            $this->searchFields[$k]='';
        }else{
            $this->searchFields[$k]=($resource[$k] !=
            '' ? urldecode($resource[$k]) : $this->defaultSearchFields[$k]);
        }
    }
}

/**
 * @desc merge getvars from pagination and searchfields
 */
private function getGetVars()
{
    return isset($_GET['page']) ?
    array_merge(array('page'=>$_GET['page']), $this->searchFields) :
    $this->searchFields;
}

/**
 * @desc addassignments
 */
public function actionManage(){

    // get changable columnnames
    $colUsername=Yii::app()->controller->module-
>columnUsername;
    $colUserid=Yii::app()->controller->module-
>columnUserid;

    // check access to view
    $this->checkAccess('RbacAssignmentViewer', true);
    if(isset($_GET['userid']))
    {

```

```

        // warn if user is protected
        if(in_array($_GET['userid'], $this->protectedUsers))
            $this->messageWarnings[]="Warning! User is protected by Controller";

        // user must exist
        if($user=User::model()->findByAttributes(array("$colUserId"=>urldecode($_GET['userid'])))
        ){
            $this->manageUser=$user;
        }else{
            throw new CHttpException("Selected User ".urldecode($_GET['username'])." does not exist");
        }
        }elseif(isset($_POST['userid']))
        {
            // check access for edit assignments
            $this->checkAccess('RbacAssignmentEditor', true);

            if(in_array($_POST['userid'], $this->protectedUsers))
            {
                $this->messageErrors[]="Sorry, User is protected by Controller";
                $this->actionIndex();
            }
            $username = $_POST['username'];
            $userid=(int)$_POST['userid'];
            if(!$user=User::model()->findByAttributes(array("$colUserId"=>$userid)))
                throw new CHttpException("Managed User $username does not exist");
            // add selected assignments
            if(isset($_POST['addAssignments']))
            {
                // fill bizRule with deny-always code if selected from user
                $bizRule=isset($_POST['secureMode']) ? 'return false;' : '';
                foreach($_POST['addAssignments'] as $itemname)
                {
                    // add default code to bizRule if selected
                    if(isset($_POST['addData'])) {
                        $item=AuthItem::model()->findByAttributes(array('name'=>$itemname));
                        $bizRule.=$item->data;
                    }
                    // add assignment
                    $assignment=new AuthAssignment;
                    $assignment->attributes=(array('userid'=>$userid, 'itemname'=>$itemname, 'bizrule'=>$bizRule, 'data'=>''));
                    if(!$assignment->validate())

```



```

throw new CHttpException("New
Assignment validation Error");
$assignment->save();
$this->messageSuccess[]="Assignment
$itemname succesfull added.";
    }
    // remove selected assignments
    if(isset($_POST['removeAssignments']))
    {
        foreach($_POST['removeAssignments'] as
$itemname)
        {
            $assignment=AuthAssignment::model()-
>findByAttributes(array('userid'=>$userid,
'itemname'=>$itemname));
            $assignment->delete();
            $this->messageSuccess[]="Assignment
$itemname succesfull removed.";
        }
        $this->manageUser=$user;
    }else{
        $this->actionIndex();
    }
    $this->manageUser=$user;
    $this->_getSearchFields();

    $displayHelper=new RBACDisplayHelper($this,
'renderItemAssign');
    $displayHelper->setUser($this->manageUser);

    $this->doRender('manage', array(
        'displayHelper'=>$displayHelper,
        'manageUser'=>$this->manageUser,
        'getVars' => $this->getGetVars()
    ));
    Yii::app()->end();
}

/**
 * @desc eject assignment or edit assignment data
 */
public function actionEdit()
{
    $colUsername=Yii::app()->controller->module-
>columnUsername;
    $colUserId=Yii::app()->controller->module-
>columnUserId;

    $this->checkAccess('RbacAssignmentViewer', true);

    if(isset($_GET['userid']))
    {
        if(in_array($_GET['userid'], $this-
>protectedUsers))

```

```

        {
            $this->messageWarnings[]="Warning! User is
protected by Controller";
        }
        if($this->editUser=User::model()-
>findByAttributes(array($colUserid=>$_GET['userid']))
        {
            $this-
>assignments=AuthAssignment::model()-
>findAllByAttributes(array('userid'=>$this->editUser-
>$colUserid));

            $this->_getSearchFields();
            $this->doRender('edit', array(
                'user'=>$this->editUser,
                'assignments'=>$this->assignments,
                'getVars' => $this->getGetVars()
            ));
        }else{
            throw new CHttpException("Selected User
".CHtml::encode($_GET['username'])." does not exist");
        }
        }elseif(isset($_POST['userid'])
            && isset($_POST['assignments'])
            && is_array($_POST['assignments']))
        {
            $this->checkAccess('RbacAssignmentEditor',
true);
            if(in_array($_POST['userid'], $this-
>protectedUsers))
            {
                $this->messageErrors[]="Sorry, User is
protected by Controller";
                $this->actionIndex();
            }
            if($this->editUser=User::model()-
>findByAttributes(array($colUserid=>$_POST['userid']))
            {
                foreach($_POST['assignments'] as $itemName
=> $values)
                {

                    $modelAssign=AuthAssignment::model()-
>findByAttributes(array('itemName'=>$itemName, 'userid'=>$this-
>editUser->$colUserid));

                    $modelAssign-
>attributes=array('bizrule'=>$values['bizrule'],
'data'=>$values['data']);

                    if($modelAssign->validate())
                    {
                        $modelAssign->save();
                        $this-
>messageSuccess[]="Assignment $itemName successfull updated.";
                    }
                }
            }
            $this->_getSearchFields();
            $this->doRender('edit', array(

```

```

        'user'=>$this->editUser,
        'assignments'=>$this->
>assignments=AuthAssignment::model()-
>findAllByAttributes(array('userid'=>$this->editUser-
>$colUserId)),
        'getVars' => $this->
>getGetVars()
    ));
    Yii::app()->end();
}
    }else{
        throw new CHttpException("User
".CHtml::encode($_POST['username'])." does not exist");
    }
    }else{
        throw new CHttpException("Not enough Data for
Edit Assignments found");
    }
}

}
?>

```

RbacController.php

```

<?php

class RbacController extends RBACBaseController
{
    public $rejectItemsGet=array();
    public $rejectItemsPost=array();

    public $editItem;
    public $authItem;

    public $filterNames='/^[a-zA-Z]+[a-zA-Z0-9-_]+/';
    /*
     * @desc this is called by default and after each action of
this controller
     */
    public function actionIndex()
    {
        $this->checkAccess('RbacViewer', true);
        $model=new RBACTree;
        if(isset($_GET['createNew']))
            $this->editItem=AuthItem::model();
        $treeFormat = 'renderItemTree';
        if(isset($_GET['generateCode']))
        {
            $treeFormat = 'renderCode';
        }elseif(isset($_GET['item']) ||
isset($_GET['createNew']) || (!empty($_POST) && count($this->
messageErrors))) {
            $treeFormat = 'renderItemInfo';
        }

        $this->doRender('index', array(

```

```

        'model'=>$this,
        'displayHelper'=>new RBACDisplayHelper($this,
$treeFormat),
        'treeFormat'=>$treeFormat,
        'ejectItemsGet'=>$this->ejectItemsGet,
        'ejectItemsPost'=>$this->ejectItemsPost,
        'editItem'=>$this->editItem,
        'authItem'=>$this->authItem,
        'nameFilter'=>$this->filterNames
    )
    );
    Yii::app()->end();
}

/**
 *
 * @desc
 */
public function actionEdit()
{
    $this->checkAccess('RbacViewer', true);
    $model=new AuthItem;
    if(empty($_POST)){
        if(isset($_GET['item'])){
            {
                if(in_array($_GET['item'], $this-
>protectedItems))
                {
                    $this->messageErrors[]="Warning!
Item is protected by Controller";
                }
                $model->attributes=$_GET;
                if($model->validate()){
                    $name=urldecode($_GET['item']);
                    if($item=$model-
>findByAttributes(array('name'=>$name)){
                        // display edit Item box
                        $this->editItem=$item;
                        $this->actionIndex();
                    }else{
                        $this->messageErrors[]="The
Item you want to edit does not exist";
                    }
                }else{
                    $this->messageErrors[]="Unsecure
Data detected. Please mail the Siteadmin if this Problem
returns.";
                }
            }
        }else{
            //ignore missing item and display index
            $this->actionIndex();
        }
    }else{
        $this->checkAccess('RbacEditor', true);
    }
}

```

```

        // filter names
        $_POST['editItem']['name']=$this-
>filterString($_POST['editItem']['name'], $this->filterNames);
        $model->attributes=$_POST['editItem'];
        $oldName=$_POST['oldName'];
        if(in_array($oldName, $this->protectedItems) ||
in_array($_POST['editItem']['name'], $this->protectedItems))
        {
            $this->messageErrors[]="Sorry, Item is
protected by Controller";
            $this->actionIndex();
        }
        if($model->validate()){
            if(isset($_POST['updateItem']))
            {
                $this-
>_updateItem($_POST['editItem'], $oldName);
            }elseif(isset($_POST['createItem']))
            {
                if(!AuthItem::model()-
>findByAttributes(array('name'=>$_POST['editItem']['name'])))
                {
                    $model->setIsNewRecord(true);
                    $model->save();
                    $this->messageSuccess[]="Item
{$_POST['editItem']['name']} successfull created.";
                }else{
                    $this->messageErrors[]="Create
Error: New Item <i>{$_POST['editItem']['name']}</i> already
exists";
                    $this->editItem=$model;
                    $this->actionIndex();
                }
            }elseif(isset($_POST['deleteItem']))
            {
                AuthItem::model()-
>deleteAllByAttributes(array('name'=>$oldName));
                AuthItemChild::model()-
>deleteAllByAttributes(array('parent'=>$oldName));
                AuthItemChild::model()-
>deleteAllByAttributes(array('child'=>$oldName));
                AuthAssignment::model()-
>deleteAllByAttributes(array('itemname'=>$oldName));
                $this->messageSuccess[]="Item
$oldName successfull deleted.";
            }else
            {
                // ignore not existing submit option
and render page
            }
            $this->actionIndex();
        }else{
            //use Yii error system
            $model->setIsNewRecord(true);
            $this->editItem=$model;
            $this->actionIndex();
        }
    }
}

```

```

    }
}

/**
 *
 * @desc
 * @param unknown_type $model
 * @param unknown_type $attributes
 * @param unknown_type $oldName
 */
private function _updateItem($attributes, $oldName)
{
    if(in_array($oldName, $this->protectedItems) ||
in_array($attributes['name'], $this->protectedItems))
    {
        $this->messageErrors[]="Sorry, Item is protected
by Controller";
        $this->actionIndex();
    }
    if(!$item=AuthItem::model()-
>findByAttributes(array('name' => $oldName)))
    {
        $this->messageErrors[]="Edit Error: Update Item
does not exist";
        $this->actionIndex();
    }

    if($attributes['type']==0 && $item->type>0)
    {
        if(count(AuthItemChild::model()-
>findAllByAttributes(array('parent'=>$oldName))))
        {
            $this->messageErrors[]="Type <i>Action</i>
can't have Childs.<br/>Please eject Childs from <i>$oldName</i>
before switch type to <i>Operation</i>";
            $this->editItem=$item;
            $this->actionIndex();
        }
    }

    if($attributes['name'] != $oldName)
    {
        if(AuthItem::model()-
>findByAttributes(array('name'=>$attributes['name'])))
        {
            $this->messageErrors[]="Create Error: New
Item <i>{$_POST['editItem']['name']}</i> already exists";
            //
            return;
        }
        $item->attributes=$attributes;
        $item->save();
        // update RBAC-Tree AuthItemChild bindings in
parent

```

```

        $newName=$attributes['name'];
        $treeItems=AuthItemChild::model()-
>findAllByAttributes(array('parent'=>$oldName));
        foreach($treeItems as $treeItem)
        {
            $treeItem->parent=$newName;
            $treeItem->save();
        }
        // update RBAC-Tree AuthItemChild bindings in
child
        $treeItems=AuthItemChild::model()-
>findAllByAttributes(array('child'=>$oldName));
        foreach($treeItems as $treeItem)
        {
            $treeItem->child=$newName;
            $treeItem->save();
        }
        // update AuthAssignment bindings in itemname
        $assignments=AuthAssignment::model()-
>findAllByAttributes(array('itemname'=>$oldName));
        foreach($assignments as $assignment)
        {
            $assignment->itemname=$newName;
            $assignment->save();
        }
    }else{
        // simple update if primary key is same
        $item->attributes=$attributes;
        $item->save();
    }
    $this->messageSuccess[]="Item ". (!isset($newName) ?
SoldName : $newName)." successfull updated.";
}

/**
 * @desc collect _POST and _GET Data, compare and validate-
>required them.
 * @desc on fail abort eject and actionIndex
 * @return array ('parent'=>string, 'child'=>string)
 */
private function validateEject()
{
    // collect vars
    if(empty($_POST))
    {
        if(!isset($_GET['parent'])) {
            $this->messageErrors[]="Eject Error: Item
".$_GET['child']."' is already ejected or in top Level";
            $this->actionIndex();
        }
    }

    $itemNames=array('parent'=>urlencode($_GET['parent']),
'child'=>urlencode($_GET['child']));
    }else{

```

```

        $itemNames=array('parent'=>urlencode($_POST['eject']['parent
        ']), 'child'=>urlencode($_POST['eject']['child']));
    }
    $model=new AuthItemChild;
    $model->attributes=$itemNames;
    if(!$model->validate())
    {
        $this->messageErrors[]="Unsecure Data detected.
Please mail the Siteadmin if this Problem returns.";
        $this->actionIndex();
    }
    return $itemNames;
}

/**
 *
 */
public function actionEject()
{
    $this->checkAccess('RbacViewer', true);
    $itemNames=$this->validateEject();
    $model=new AuthItem();
    // check only if parent is protected
    if(in_array($itemNames['parent'], $this-
>protectedItems))
    {
        if(in_array($itemNames['child'], $this-
>protectedItems))
        {
            $this->messageErrors[]="Warning! Item is
protected by Controller";
        }
    }
    $child=$model-
>findByAttributes(array('name'=>$itemNames['child']));
    $parent=$model-
>findByAttributes(array('name'=>$itemNames['parent']));

    if(empty($_POST)){
        if(!$child || !$parent){
            $this->messageErrors[]="One ore more
Item(s) does not exist";
            $this->actionIndex();
        }else{
            $this-
>ejectItemsGet=array('parent'=>$parent, 'child'=>$child);
        }
    }else{
        $this->checkAccess('RbacEditor', true);
        $model=AuthItemChild::model();
        $model->attributes=array('parent'=>$parent-
>name, 'child'=>$child->name);
        $model->delete();
        $this->messageSuccess[]="Item {$child->name}
successfull ejected from {$parent->name}.";
    }
}

```



```

    }
    $this->actionIndex();
}

/**
 *
 * @desc
 */
public function actionMove()
{
    $this->checkAccess('RbacViewer', true);
    if(!empty($_POST))
    {
        $this->checkAccess('RbacEditor', true);
        $from=isset($_POST['moveFromItem']) ?
$_POST['moveFromItem'] : null;
        $to=isset($_POST['moveToItem']) ?
$_POST['moveToItem'] : null;

        // check only if parent is protected
        if(in_array($to, $this->protectedItems))
        {
            if(in_array($from, $this->protectedItems))
            {
                $this->messageErrors[]="Sorry, Item
is protected by Controller";
                $this->actionIndex();
            }
        }
        if(!$from || !$to || $from==$to)
        {
            $this->messageErrors[]="Please select
Parent- and Childitem and care that they are not same.";
            $this->actionIndex();
        }
        // default validate
        $model=new AuthItemChild;
        $model->attributes=array('child'=>$from,
'parent'=>$to);
        if(!$model->validate())
        {
            $this->messageErrors[]="Post validation
Error. Please mail Siteadmin if this Error returns.";
            $this->actionIndex();
        }
        // check if branch already exists
        if($model-
>findByAttributes(array('child'=>$from, 'parent'=>$to)) !== null)
        {
            $this->messageErrors[]="Create Branch
Error: Branch already exists.";
            $this->actionIndex();
        }
    }
}

```

```

        // Items exist?
        $model=new AuthItem;
        if(!count($model->findByAttributes(array('name'=>$from))) || !count($model->findByAttributes(array('name'=>$to))))
        {
            $this->messageErrors[]="Check Items exists Error. Please mail Siteadmin if this Error returns.";
            $this->actionIndex();
        }
        // make recursioncheck and move Items
        $model=new RBACTree;
        $model->moveFrom=$from;
        $model->moveTo=$to;
        if($model->checkRecursion())
        {
            $model->moveItem();
            $this->messageSuccess[]="Item $from successfull moved to $to.";
        }else{
            $this->messageErrors[]="Can't move Selection cause that will produce a Recursion.
            <br>If you can't see producing a Recursion, it may help to eject Item before moving it.";
            $this->actionIndex();
        }
    }
    $this->actionIndex();
}
}

```

7. Pesan

a. Model

Mailbox.php

```
<?php
```

```

/**
 * This is the model class for table "{{mailbox_conversation}}".
 *
 * The followings are the available columns in table '{{mailbox_conversation}}':
 * @property string $conversation_id
 * @property integer $initiator_id
 * @property integer $interlocutor_id
 * @property string $subject
 * @property integer $bm_read
 * @property integer $bm_deleted
 * @property string $modified
 * @property string $is_system
 */
class Mailbox extends CActiveRecord
{
    const INITIATOR_FLAG = 1;
    const INTERLOCUTOR_FLAG = 2;
    public $to;
}

```

```

public $text;
public $sender_id;
public $is_replied;

/**
 * Returns the static model of the specified AR class.
 * @param string $className active record class name.
 * @return Mailbox the static model class
 */
public static function model($className=__CLASS__)
{
    return parent::model($className);
}

/**
 * @return string the associated database table name
 */
public function tableName()
{
    return MailboxModule::TBL_CONV;
}

/**
 * @return array validation rules for model attributes.
 */
public function rules()
{
    // NOTE: you should only define rules for those
attributes that
    // will receive user inputs.
    return array(
        //array('initiator_del,interlocutor_del',
'default', 'setOnEmpty'=>true, 'value'=>null ),
        array('initiator_id, interlocutor_id, modified',
'required'),
        array('initiator_id, interlocutor_id, bm_read,
bm_deleted, initiator_del, interlocutor_del', 'numerical',
'integerOnly'=>true),
        array('subject', 'length', 'max'=>100),
        array('modified', 'length', 'max'=>10),
        array('is_system', 'length', 'max'=>3),

        array('subject','match','pattern'=>' /^[^'.Yii::app()-
>controller->module->allowableCharsSubject.']+ /i','not'=>true),
        // The following rule is used by search().
        // Please remove those attributes that should
not be searched.
        array('conversation_id, initiator_id,
interlocutor_id, subject, bm_read, bm_deleted, modified,
is_system', 'safe', 'on'=>'search'),
        array('to','length','max'=>30),
    );
}

/**
 * @return array relational rules.

```

```

        */
        public function relations()
        {
            // NOTE: you may need to adjust the relation name and
            the related
            // class name for the relations automatically
            generated below.
            return array(
                'initiator'=>array(self::BELONGS_TO,
                'User','initiator_id'),
                'interlocutor'=>array(self::BELONGS_TO,
                'User','interlocutor_id'),
                'messages'=>array(self::HAS_MANY ,
                'Message','conversation_id'),
            );
        }

        /**
         * @return array customized attribute labels (name=>label)
         */
        public function attributeLabels()
        {
            return array(
                'conversation_id' => 'Id',
                'initiator_id' => 'From Id',
                'interlocutor_id' => 'To Id',
                'to' => 'To',
                'subject' => 'Subject',
                'last_message_ts' => 'Last Message Received',
                'modified' => 'Last Modified',
                'is_system' => 'Is System',
            );
        }

        /**
         * Retrieves a list of models based on the current
         search/filter conditions.
         * @return CActiveDataProvider the data provider that can
         return the models based on the search/filter conditions.
         */
        public function search()
        {
            // Warning: Please modify the following code to remove
            attributes that
            // should not be searched.

            $criteria=new CDbCriteria;

            $criteria->compare('conversation_id',$this->conversation_id,true);
            $criteria->compare('initiator_id',$this->initiator_id);
            $criteria->compare('interlocutor_id',$this->interlocutor_id);
            $criteria->compare('subject',$this->subject,true);
            $criteria->compare('bm_read',$this->bm_read);

```

```

$criteria->compare('bm_deleted',$this->bm_deleted);
$criteria->compare('modified',$this->modified,true);
$criteria->compare('is_system',$this->is_system,true);

return new CActiveDataProvider($this, array(
    'criteria'=>$criteria,
));
}

public function trash($userid)
{
    $this->getDbCriteria()->mergeWith(array(
        'select'=>'m.message_id, m.conversation_id as
conversation, COUNT(m.message_id) AS mails_count, m.sender_id,
m.recipient_id,
        m.text, m.created AS last_message_ts,
ms.is_replied,
        c.conversation_id, c.initiator_id,
c.interlocutor_id, c.subject, c.bm_read,
c.bm_deleted, c.modified, c.is_system',
        'alias'=>'c',
        'join'=>"INNER JOIN (
            SELECT message_id,conversation_id,
sender_id,recipient_id,text,created FROM
".MailboxModule::TBL_MSG."
            WHERE sender_id=:userid OR
recipient_id=:userid ORDER BY created DESC
        ) AS m ON(m.conversation_id=c.conversation_id)
        INNER JOIN (
            SELECT conversation_id,
IF(sender_id=:userid,1,0) AS is_replied FROM
".MailboxModule::TBL_MSG."
            WHERE (recipient_id=:userid OR
sender_id=:userid) ORDER BY created DESC
        ) AS ms
ON(ms.conversation_id=c.conversation_id)",
        'condition'=>'(initiator_id=:userid OR
interlocutor_id=:userid) AND
IF(initiator_id=:userid,initiator_del,interlocutor_del) > 0',
        'group'=>"c.conversation_id",
        'order'=>"MAX( m.created ) DESC",
        'params'=>array(':userid'=>$userid,

':bminit'=>self::getBitMaskValues(self::INITIATOR_FLAG),

':bminter'=>self::getBitMaskValues(self::INTERLOCUTOR_FLAG)),
    ));
    return $this;
}

public function inbox($userid)
{
    $this->getDbCriteria()->mergeWith(array(
        'select'=>'
        m.message_id, m.conversation_id as conversation,

```

```

COUNT(m.message_id) AS mails_count, m.sender_id,
m.recipient_id,
m.text, m.created AS last_message_ts,
ms.is_replied,
c.conversation_id, c.initiator_id,
c.interlocutor_id, c.subject, c.bm_read,
c.bm_deleted, c.modified, c.is_system',
'alias'=>'c',
'join'=>"INNER JOIN (
    SELECT message_id, conversation_id,
sender_id, recipient_id, text, created FROM
".MailboxModule::TBL_MSG."
    WHERE recipient_id=:userid ORDER BY
created DESC
    ) AS m ON(m.conversation_id=c.conversation_id)
    INNER JOIN (
        SELECT conversation_id,
IF(sender_id=:userid,1,0) AS is_replied FROM
".MailboxModule::TBL_MSG."
        WHERE (recipient_id=:userid OR
sender_id=:userid) ORDER BY created DESC
    ) AS ms
ON(ms.conversation_id=c.conversation_id)",
'condition'=>'(c.initiator_id=:userid OR
c.interlocutor_id=:userid) AND (c.bm_deleted &
IF(c.initiator_id=:userid,:bminit,:bminter)) = 0',
'group'=>"c.conversation_id",
'order'=>"MAX( m.created ) DESC",
'params'=>array(':userid'=>$userid,
':bminit'=>self::INITIATOR_FLAG,
':bminter'=>self::INTERLOCUTOR_FLAG),
));
return $this;
}

public function conversations($userid,$type='deleted')
{
    switch ($type) {
        case 'deleted':
            $where = "c.initiator_del > 0 OR
c.interlocator_del > 0";
            break;
        default: $where = 'c.bm_deleted != 3';
    }
    $query = "SELECT m.message_id, m.conversation_id,
COUNT(m.message_id) AS mails_count, m.sender_id, m.recipient_id,
m.text, m.created AS last_message_ts,
ms.is_replied,
c.conversation_id, c.initiator_id,
c.interlocutor_id, c.subject, c.bm_read,
c.bm_deleted, c.modified, c.is_system
FROM ".MailboxModule::TBL_CONV." AS c
WHERE $where";

    $sql = Yii::app()->db->createCommand($query);

```

```

        $sql->bindValue(':userid',$userid,PDO::PARAM_STR);
        $sql-
>bindValue(':bminit',self::getBitMaskValues(self::INITIATOR_FLAG),
PDO::PARAM_STR);
        $sql-
>bindValue(':bminter',self::getBitMaskValues(self::INTERLOCUTOR_FL
AG),PDO::PARAM_STR);
        //$sql->bindParam(':page',$page,PDO::PARAM_INT);
        //$sql->bindParam(':perpage',$perpage,PDO::PARAM_INT);
        $convs = $sql->queryAll();

        // or using: $rawData=User::model()->findAll();
        $dataProvider=new CArrayDataProvider($convs, array(
            'keyField'=>'conversation_id',
            'id'=>'inboxDataProvider',
            'sort'=>array(
                'attributes'=>array(
                    'subject', 'last_message_ts',
'sender_id',
                    // 'mails_count','initiator_id',
'interlocutor_id', 'recipient_id', 'text',
                    // 'bm_read','bm_deleted', 'is_system',
'is_readable', 'is_replied', 'conversation_id',
                ),
            ),
            'pagination'=>array(
                'pageSize'=>Yii::app()->controller-
>module->pageSize,
            ),
        ));

        return $dataProvider;
    }

    public static function newMsgs($userid)
    {
        // count messages
        $query = "SELECT SQL_CACHE COUNT(c.conversation_id) AS
num_messages FROM
        ".MailboxModule::TBL_CONV." AS c
        INNER JOIN ( SELECT message_id,conversation_id
        FROM ".MailboxModule::TBL_MSG."
        WHERE recipient_id=:userid
        ORDER BY created DESC
        ) AS m
ON(m.conversation_id=c.conversation_id)
        WHERE (c.initiator_id=:userid OR
c.interlocutor_id=:userid)
        AND (c.bm_read & IF(c.initiator_id=:userid,
:bminit, :bminter)) = 0
        AND (c.bm_deleted & IF(c.initiator_id=:userid,
:bminit, :bminter)) = 0
        GROUP BY c.conversation_id
        ";
        $sql = Yii::app()->db->createCommand($query);
        $sql->bindValue(':userid',$userid,PDO::PARAM_STR);

```

```

        $sql-
>bindValue(':bminit',self::INITIATOR_FLAG,PDO::PARAM_STR);
        $sql-
>bindValue(':bminter',self::INTERLOCUTOR_FLAG,PDO::PARAM_STR);

        $convs = $sql->queryAll();
        $count=0;
        foreach($convs as &$conv) {
            if ($conv['num_messages'] >= 1)
                $count++;
        }
        return $count;
    }

    public function deleted()
    {
        $this->getDbCriteria()->mergeWith(array(
            'select'=>'conversation_id, initiator_id,
interlocutor_id, c.bm_deleted, c.modified, c.is_system',
            'condition'=>'c.initiator_del > 0 OR
c.interlocutor_del > 0',
        ));
        return $this;
    }

    public static function conversation($id,$order='DESC')
    {
        if(!is_int((int)$id))
            throw new Exception('Bad conversation Id');
        $with = array('messages'=>array('order'=>'created
'.'. $order));
        return Mailbox::model()->with($with)->findByPk($id);
    }

    public function markRead($userid)
    {
        if($this->initiator_id == $userid)
            $this->bm_read = $this->bm_read |
Mailbox::INITIATOR_FLAG;
        else
            $this->bm_read = $this->bm_read |
Mailbox::INTERLOCUTOR_FLAG;
        $this->updateByPk($this-
>conversation_id,array('bm_read'=>$this->bm_read));
    }

    public function belongsTo($userid)
    {
        return ($this->initiator_id == $userid || $this-
>interlocutor_id == $userid);
    }

    public function isNew($userid)

```



```

    {
        if($this->initiator_id == $userid )
            $flag = self::INITIATOR_FLAG;
        else
            $flag = self::INTERLOCUTOR_FLAG;

        return !($this->bm_read & $flag);
    }

    /*
     * Button Actions
     */
    public function read($userid)
    {
        if($this->initiator_id == $userid) {
            $this->bm_read = $this->bm_read |
self::INITIATOR_FLAG;
        }
        elseif($this->interlocutor_id == $userid) {
            $this->bm_read = $this->bm_read |
self::INTERLOCUTOR_FLAG;
        }
        else
            throw new Exception('User denied');
        return true;
    }

    public function unread($userid)
    {
        if($this->initiator_id == $userid) {
            $this->bm_read = $this->bm_read &
~self::INITIATOR_FLAG;
        }
        elseif($this->interlocutor_id == $userid) {
            $this->bm_read = $this->bm_read &
~self::INTERLOCUTOR_FLAG;
        }
        else
            throw new Exception('User denied');
        return true;
    }

    public function restore($userid)
    {
        if($this->initiator_id == $userid) {
            $this->bm_deleted = $this->bm_deleted &
~self::INITIATOR_FLAG;
            $this->initiator_del = 0;
        }
        elseif($this->interlocutor_id == $userid) {
            $this->bm_deleted = $this->bm_deleted &
~self::INTERLOCUTOR_FLAG;
            $this->interlocutor_del = 0;
        }
        else throw new Exception('User denied');
    }

```

```

        return true;
    }

    /**
     * Delete the conversation in respect to user's perspective.
     When both initiator and interlocutor have
     * permanently deleted the conversation, only then is the
     conversation actually removed from the
     * database. If 'trashbox' is enabled in the module configs
     then instead of marking the conversation
     * as permanently deleted by that user, the conversation is
     moved to the trash folder by setting
     * {perspective}_del field to the day of the half-year
     "date('z') / 2" that the message was deleted.
     * Also if 'trashbox' is enabled then the cron method in
     MailboxModule must be ran at least once
     * daily in order to check the number of days the message
     has been in the trash folder, and delete it
     * permanently if days is more than config var
     'recyclePeriod'.
     *
     * @param integer $userid the user's Id in which to delete
     the conversation in respect to. The parameter
     * declaration is given a default value in order to overload
     the parents delete method and ensure that
     * conversations are deleted via the destroy method.
     * @return boolean true on success
     */
    public function delete($userid=0)
    {
        //die((int)Yii::app()->controller->module->
        >recyclePeriod);
        if(!$userid)
            throw new Exception('User Id must be supplied
            for delete method');
        $day = date("z") + 1;
        if($day > 183)
            $day = $day - 183;
        if($this->initiator_id == $userid) {
            // mark deleted
            $this->bm_deleted = $this->bm_deleted |
            self::INITIATOR_FLAG;
            // check if in trash already, or if trashbox is
            disabled
            if(!$this->initiator_del && $this->
            >initiator_del > Yii::app()->controller->module->recyclePeriod)
                || !Yii::app()->controller->module->
            >trashbox) {
                // permanently delete
                $this->initiator_del = null;
            } else {
                // move to trash
                $this->initiator_del = $day;
            }
        }
    }
}

```

```

elseif($this->interlocutor_id == $userid) {
    // mark deleted
    $this->bm_deleted = $this->bm_deleted |
self::INTERLOCUTOR_FLAG;
    // check if in trash already, or if trashbox is
disabled
    if($this->interlocutor_del > Yii::app()-
>controller->module->recyclePeriod || !Yii::app()->controller-
>module->trashbox) {
        // permanently delete
        $this->interlocutor_del = null;
    } else {
        // move to trash
        $this->interlocutor_del = $day;
    }
}
else
    throw new Exception('User denied');

    // if both parties have permanently deleted
conversation
    if(is_null($this->initiator_del) && is_null($this-
>interlocutor_del))
    {
        // delete conversation and messages from
database
        $this->destroy();
    }
    return true;
}
/*
 * End button actions
 */

/**
 * This method is used by the cron method in the
MailboxModule class to determine when a conversation
 * should be marked as permanently deleted by a user
depending on the recycling period defined in the
 * module configs by the attribute 'recyclePeriod'. And if
both users have the conversation marked
 * as permanently deleted then remove the conversation from
the database.
 *
 * @param integer $mask
 * @param boolean $string_presentation
 * @return string|array
 */
public function recycle()
{
    // check initiator trash
    if($this->initiator_del > 0)
    {
        // if not within recycling period
        if(!self::withinDays(Yii::app()->controller-
>module->recyclePeriod,

```

```

        $this->initiator_del)) {
            // mark permanently deleted
            $this->initiator_del = null;
        }
    }
    // check interlocutor trash
    if($this->interlocutor_del > 0)
    {
        // if not within recycling period
        if(!self::withinDays(Yii::app()->controller-
>module->recyclePeriod,
            $this->interlocutor_del)) {
            // mark permanently deleted
            $this->interlocutor_del = null;
        }
    }
    // if both marked as permanently deleted
    if(is_null($this->initiator_del) && is_null($this-
>interlocutor_del))
    {
        // delete this conversation from database
        $this->destroy();
    }
    else{
        // otherwise save possible changes
        $this->save();
    }
}

public function destroy()
{
    // delete all conversation messages
    $msg_count=Message::model()-
>deleteAll('conversation_id = :cid',array(':cid'=>$this-
>conversation_id));
    if(!$msg_count)
        throw new Exception('Conversation messages not
found?');

    // delete conversation
    return parent::delete();
}

/**
 * Get array of values that make up mask.
 *
 * @param integer $mask
 * @param boolean $string_presentation
 * @return string|array
 */
public static function getBitMaskValues( $mask,
$string_presentation = true )
{
    $available_mask_arr = array(self::INITIATOR_FLAG,
self::INTERLOCUTOR_FLAG);

```

```

        if ( in_array( $mask, $available_mask_arr ) )
        {
            $values_count = pow( 2, 2 );
            if ( $string_presentation )
            {
                $return_arr = "";
                for ( $i = 1; $i < $values_count; $i++ )
                    if ( ($mask & $i) == $mask )
                        $return_arr .= $i.", ";

                return substr( $return_arr, 0, -2 );
            }
            else
            {
                $return_arr = array();
                for ( $i = 1; $i < $values_count; $i++ )
                    if ( ($mask & $i) == $mask )
                        $return_arr[$i] = $i;
                return $return_arr;
            }
        }
    }

    public static function withinDays($period,$deleted_on)
    {
        $day = date("z") + 1;
        if($day > 183)
            $day = $day - 183;
        elseif($day < $deleted_on)
            $day += 183;

        if( ($day - $deleted_on) <= $period )
            return true;

        return false;
    }
}

```

Message.php

```

<?php

/**
 * This is the model class for table "{{mailbox_message}}".
 *
 * The followings are the available columns in table
 * '{{mailbox_message}}':
 * @property string $message_id
 * @property string $conversation_id
 * @property string $created
 * @property string $sender_id
 * @property string $recipient_id
 * @property string $text
 * @property string $is_kiss
 * @property string $is_readable
 * @property string $hash

```

```

*/
class Message extends CActiveRecord
{
    public $modified;
    public $subject;
    public $is_replied;
    public $initiator_id;
    public $bm_read;
    /**
     * Returns the static model of the specified AR class.
     * @param string $className active record class name.
     * @return Message the static model class
     */
    public static function model($className=__CLASS__)
    {
        return parent::model($className);
    }

    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return MailboxModule::TBL_MSG;
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        $module =& Yii::app()->controller->module;
        $p = new CHtmlPurifier();
        if(Yii::app()->controller->module->isAdmin())
            $p->options = $module->adminHtml;
        else
            $p->options = $module->html;
        return array(
            array('text', 'required'),
            array('created, sender_id, recipient_id',
                'length', 'max'=>10),

            array('text', 'filter', 'filter'=>array($p, 'purify')),
            array('file_foto', 'file', 'types'=>'doc, docx,
pdf, rar, zip, 7z', 'allowEmpty' => true),
            // The following rule is used by search().
            // Please remove those attributes that should
            not be searched.
            array('message_id, conversation_id, created,
sender_id, recipient_id, text, hash, file_foto', 'safe',
            'on'=>'search'),
        );
    }

    /**
     * @return array relational rules.

```

```

        */
        public function relations()
        {
            // NOTE: you may need to adjust the relation name and
            the related
            // class name for the relations automatically
            generated below.
            return array(
                'conversation'=>array(self::HAS_ONE,
'Mailbox','conversation_id'),
            );
        }

        /**
         * @return array customized attribute labels (name=>label)
         */
        public function attributeLabels()
        {
            return array(
                'message_id' => 'Message',
                'conversation_id' => 'Conversation Id',
                'created' => 'Time Stamp',
                'sender_id' => 'Sender',
                'recipient_id' => 'Recipient',
                'text' => 'Text',
                'file_foto' => 'File',
            );
        }

        /**
         * Retrieves a list of models based on the current
         search/filter conditions.
         * @return CActiveDataProvider the data provider that can
         return the models based on the search/filter conditions.
         */
        public function search()
        {
            // Warning: Please modify the following code to remove
            attributes that
            // should not be searched.

            $criteria=new CDbCriteria;

            $criteria->compare('message_id',$this-
>message_id,true);
            $criteria->compare('conversation_id',$this-
>conversation_id,true);
            $criteria->compare('created',$this->created,true);
            $criteria->compare('sender_id',$this->sender_id,true);
            $criteria->compare('recipient_id',$this-
>recipient_id,true);
            $criteria->compare('text',$this->text,true);
            $criteria->compare('file_foto',$this->file_foto,true);

            return new CActiveDataProvider($this, array(
                'criteria'=>$criteria,
            ));
        }
    }
}

```

```

        ));
    }

    public function conversation($conversation_id)
    {
        if(!preg_match('/^[0-9]+$/', $conversation_id))
            die('Conversation Id must be an
integer:'. $conversation_id);
        $this->getDbCriteria()->mergeWith(array(
            'select'=>'*',
            'condition'=>'conversation_id = :cid',
            'params'=>array(':cid'=>$conversation_id)
        ));
        return $this;
    }

    public function sent($userid)
    {
        $this->getDbCriteria()->mergeWith(array(
            'select'=>'m.message_id, m.sender_id,
m.recipient_id,
            m.text, m.created,
            c.conversation_id, c.initiator_id ,
c.interlocutor_id, c.subject, c.bm_read,
            c.bm_deleted, c.modified, c.is_system,
            ms.created as is_replied',
            'alias'=>'m',
            'join'=>"
            INNER JOIN ".MailboxModule::TBL_CONV." AS c
ON(c.conversation_id=m.conversation_id)
            LEFT JOIN (
                SELECT conversation_id, created FROM
".MailboxModule::TBL_MSG."
                WHERE recipient_id=:userid AND created
            ) AS ms ON(ms.conversation_id=m.conversation_id
AND ms.created > m.created)
            ",
            'condition'=>'sender_id=:userid',
            'order'=>"m.created DESC",
            'params'=>array(':userid'=>$userid),
        ));
        return $this;

        /* old */
        $is_replied = "
            INNER JOIN (
                SELECT conversation_id,
IF(sender_id=:userid,'yes','no') AS is_replied FROM
".MailboxModule::TBL_MSG."
                WHERE (recipient_id=:userid OR
sender_id=:userid) ORDER BY created DESC
            ) AS ms
ON(ms.conversation_id=c.conversation_id)";
    }

```



```

        public static function crc64($text)
        {
            $crc64 = sprintf('%u',hash('crc32', $text)) .
sprintf('%u',hash('crc32b', $text));
            return base_convert($crc64,16,10); // 64bit INT
        }

        public function isRead($userid)
        {
            if($this->is_replied)
                return 1;
            if($this->initiator_id == Yii::app()->controller-
>module->getUserId() )
                $flag = Mailbox::INTERLOCUTOR_FLAG;
            else
                $flag = Mailbox::INITIATOR_FLAG;

            return $this->bm_read & $flag;
        }
    }
}

```

News.php

```

<?php

/**
 * This is the model class for table "{{mailbox_conversation}}".
 *
 * The followings are the available columns in table
 '{{mailbox_conversation}}':
 * @property string $conversation_id
 * @property integer $initiator_id
 * @property integer $interlocutor_id
 * @property string $subject
 * @property integer $bm_read
 * @property integer $bm_deleted
 * @property string $modified
 * @property string $is_system
 */
class News extends Mailbox
{
    public static function model($className=__CLASS__)
    {
        return parent::model($className);
    }

    public function isNew($userid)
    {
        /*
         * If your database keeps track of user logins
         date/time you could rewrite
         * this method to return true if the modified
         timestamp is greater than
         * (ie. more recent) the user's previous login
         timestamp (ie the last time

```

```

        * the user successfully logged-in, not counting the
current login).
        */
        /*
        if($this->modified > Yii::app()->user->previousLogin )
            return true;
        return false;
        */

        // for now we just use a static time limit...
        if($this->modified > time() - 3600 * 24 * 7)
            return true;
        return false;
    }

    public function delete()
    {
        return $this->destroy();
    }
}

```

b. View

mailbox.php

```

<?php

if($this->getAction()->getId()!='inbox')
    $this->breadcrumbs=array(
        ucfirst($this->module->id)=>array('inbox'),
        ucfirst($this->getAction()->getId())
    );
else
    $this->breadcrumbs=array($this->module->id);

$this->renderpartial('_menu');

if(isset($_GET['Message_sort']))
    $sortby = $_GET['Message_sort'];
elseif(isset($_GET['Mailbox_sort']))
    $sortby = $_GET['Mailbox_sort'];
else
    $sortby = '';
$ie6br = <<<EOD
<!--[if lt IE 6]>
<br clear="all" />
<![endif]-->
EOD;
echo '<div id="mailbox-list" class="mailbox-list ui-helper-clearfix" sortby="'.$sortby.'">';

$this->renderpartial('_flash');

if($dataProvider->getItemCount() > 0) {
?>

```

```

<form id="message-list-form" action="<?php echo $this->createUrl($this->getId().'/'.$this->getAction()->getId()); ?>"
method="post">
    <input type="hidden" class="mailbox-count" name="ui[]"
value="<?php echo $dataProvider->getItemCount(); ?>" />
    <input type="hidden" class="mailbox-sortby" name="ui[]"
value="<?php echo $sortby; ?>" />
    <div class="mailbox-clistview-container ui-helper-clearfix">
        <?php
            if($dataProvider->getItemCount() > 1 && $this->getAction()->getId() != 'sent') : ?>
                <div class="btn-group mailbox-checkall-buttons">
                    <button class="checkall" />Check All</button>
                    <button class="uncheckall" />Uncheck
                All</button>
            </div>
        <?php
            endif;

$this->widget('zii.widgets.CListView', array(
    'id'=>'mailbox',
    'dataProvider'=>$dataProvider,
    'itemView'=>'_list',
    'itemsTagName'=>'table',
    'template'=>'<div class="mailbox-
summary">{summary}</div>{sorter}'. $ie6br.'<div id="mailbox-items"
class="ui-helper-clearfix">{items}</div>{pager}',
    'sortableAttributes'=>$this->getAction()->getId()=='sent'?
        array('created'=>'Date Sent') :
        array('modified'=>'Date Received'),
    'loadingCssClass'=>'mailbox-loading',
    'ajaxUpdate'=>'mailbox-list',
    'afterAjaxUpdate'=>'$.yiimailbox.updateMailbox',
    'emptyText'=>'<div style="width:100%"><h3>You have no mail in
your '.$this->getAction()->getId().' folder.</h3></div>',
    //'htmlOptions'=>array('class'=>'ui-helper-clearfix'),
    'sorterHeader'=>'',
    'sorterCssClass'=>'mailbox-sorter',
    'itemsCssClass'=>'mailbox-items-tbl ui-helper-clearfix',
    'pagerCssClass'=>'mailbox-pager',
    //'updateSelector'=>'.inbox',
));
?>

    <?php if($this->getAction()->getId()!='sent') : ?>
<div style="clear:left"> <span class="mailbox-buttons-label">With
selected:</span>
        <?php if($this->getAction()->getId()=='trash') : ?>
            <input type="submit" id="mailbox-action-restore" class="btn
mailbox-button" name="button[restore]" value="restore" />
            <input type="submit" id="mailbox-action-delete" class="btn
mailbox-button" name="button[delete]" value="delete forever" />
        <?php else: ?>
            <?php if(!$this->module->readOnly || ( $this->module->readOnly && !$this->module->isAdmin())) : ?>
                <input type="submit" id="mailbox-action-delete" class="btn
mailbox-button" name="button[delete]" value="delete" />
            </?php
        </?php
    </div>
</form>

```

```

        <?php endif; ?>
        <input type="submit" id="mailbox-action-read" class="btn
mailbox-button" name="button[read]" value="read" />
        <input type="submit" id="mailbox-action-unread" class="btn
mailbox-button" name="button[unread]" value="unread" />
        <?php endif; ?>
</div>
        <?php endif; ?>
    </div>
</form>

<?php
}
else {
    $this->renderpartial('_empty');
}
?>
</div>

<script type="text/javascript">
/*<![CDATA[*/
jQuery(function($) {
    $('.message-subject').hide();
});
/*]]>*/
</script>

```

_mailbox.php

```

<?php

if(isset($_GET['Message_sort']))
    $sortby = $_GET['Message_sort'];
elseif(isset($_GET['Mailbox_sort']))
    $sortby = $_GET['Mailbox_sort'];
else
    $sortby = '';

echo '<div id="mailbox-list" class="mailbox-list ui-helper-
clearfix" sortby="'.$sortby.'">';

$this->renderpartial('_flash');

$ie6br = <<<EOD
<!--[if lt IE 6]>
<br clear="all" />
<![endif]-->
EOD;

if($dataProvider->getItemCount() > 0) {
?>

<form id="message-list-form" action="<?php echo $this-
>createUrl($this->getId().'/'.$this->getAction()->getId()); ?>"
method="post">

```

```

        <input type="hidden" class="mailbox-count" name="ui[]"
value="<?php echo $dataProvider->getItemCount(); ?>" />
        <input type="hidden" class="mailbox-sortby" name="ui[]"
value="<?php echo $sortby; ?>" />
        <div class="mailbox-clistview-container">
            <?php
                if($dataProvider->getItemCount() > 1 && $this->getAction()-
>getId() != 'sent') : ?>
                    <div class="btn-group mailbox-checkall-buttons">
                        <button class="btn checkall">Check All</button>
                        <button class="btn uncheckall">Uncheck
All</button>
                    </div>
                <?php
            endif;

$this->widget('zii.widgets.CListView', array(
    'id'=>'mailbox',
    'dataProvider'=>$dataProvider,
    'itemView'=>'_list',
    'itemsTagName'=>'table',
    'template'=>'<div class="mailbox-
summary">{summary}</div>{sorter}'. $ie6br. '<div id="mailbox-items"
class="ui-helper-clearfix">{items}</div>{pager}',
    'sortableAttributes'=>$this->getAction()->getId()=='sent'?
        array('created'=>'Date Sent') :
        array('modified'=>'Date Received'),
    'loadingCssClass'=>'mailbox-loading',
    'ajaxUpdate'=>'mailbox-list',
    'afterAjaxUpdate'=>'$.yiimailbox.updateMailbox',
    'emptyText'=>'<div style="width:100%"><h3>You have no mail in
your '. $this->getAction()->getId().' folder.</h3></div>',
    //'htmlOptions'=>array('class'=>'ui-helper-clearfix'),
    'sorterHeader'=>'',
    'sorterCssClass'=>'mailbox-sorter',
    'itemsCssClass'=>'mailbox-items-tbl ui-helper-clearfix',
    'pagerCssClass'=>'mailbox-pager',
    //'updateSelector'=>'.inbox',
));
?>

        <?php if($this->getAction()->getId()!='sent') : ?>
<div style="clear:left"> <span class="mailbox-buttons-label">With
selected:</span>
        <?php if($this->getAction()->getId()=='trash') : ?>
            <input type="submit" id="mailbox-action-restore" class="btn
mailbox-button" name="button[restore]" value="restore" />
            <input type="submit" class="btn" name="button[delete]"
value="delete forever" />
        <?php else: ?>
            <?php if(!$this->module->readOnly || ( $this-
>module->readOnly && !$this->module->isAdmin())) : ?>
                <input type="submit" id="mailbox-action-delete" class="btn
mailbox-button" name="button[delete]" value="delete" />
            <?php endif; ?>
            <input type="submit" id="mailbox-action-read" class="btn
mailbox-button" name="button[read]" value="read" />

```

```

        <input type="submit" id="mailbox-action-unread" class="btn
mailbox-button" name="button[unread]" value="unread" />
        <?php endif; ?>
</div>
        <?php endif; ?>
    </div>
</form>
<?php
}
else {

}
?>
</div>

<script type="text/javascript">
/**/
jQuery(function($) {
    $('.message-subject').hide();
});
/*]]&gt;*/
&lt;/script&gt;
</pre>
</div>
<div data-bbox="185 412 267 430" data-label="Section-Header">
<h3><b><u>_list.php</u></b></h3>
</div>
<div data-bbox="185 429 826 845" data-label="Text">
<pre>
&lt;?php

$userid = $this-&gt;module-&gt;getUserid();

if($this-&gt;getAction()-&gt;getId()=='sent') {
    $counterUserId = $data-&gt;recipient_id;
}
else {
    if($this-&gt;module-&gt;getUserid() == $data-&gt;initiator_id)
        $counterUserId = $data-&gt;interlocutor_id;
    else
        $counterUserId = $data-&gt;initiator_id;
}
$username = $this-&gt;module-&gt;getFromLabel($counterUserId);

if($username &amp;&amp; ($this-&gt;module-&gt;isAdmin() || $this-&gt;module-&gt;linkUser)) {
    $url = $this-&gt;module-&gt;getUrl($counterUserId);
    if($url)
        $username = '&lt;a href="'. $url. '"&gt;'. $username. '&lt;/a&gt;';
}
elseif(!$username)
    $username = '&lt;span class="mailbox-deleted-user"&gt;'. $this-&gt;module-&gt;deletedUser. '&lt;/span&gt;';

$viewLink = $this-&gt;createUrl('message/view',array('id'=&gt;$data-&gt;conversation_id));

if($this-&gt;getAction()-&gt;getId()=='sent') {
    $received = $this-&gt;module-&gt;getDate($data-&gt;created);
    if($this-&gt;module-&gt;recipientRead)
</pre>
</div>
<div data-bbox="504 923 540 939" data-label="Page-Footer">
<p>246</p>
</div>
```

```

        $itemCssClass = ($data->isRead($userid))? 'msg-read' :
'msg-deliver';
        else
            $itemCssClass = 'msg-sent';
    }
    else{
        $received = $this->module->getDate($data->modified);
        $itemCssClass = $data->isNew($userid)? 'msg-new' : 'msg-
read';
    }
    switch($itemCssClass)
    {
        case 'msg-read': $status = ($this->getAction()-
>getId()=='sent')? 'Recipient has read your message' : 'Message
has been read' ; break;
        case 'msg-deliver': $status = 'Recipient has not read your
message yet';
        case 'msg-new': $status = ($this->getAction()-
>getId()=='sent')? 'Recipient has not read your message yet' :
'You received a new message'; break;
        case 'msg-sent': $status = "You sent message {$username} a
message";
    }
    $subject = '<span class="mailbox-subject-text">';
    $subject .= '<a class="mailbox-link" title="'. $status. '"
href="'. $viewLink. '">';
    $subjectSeperator = ' - ';
    if(strlen($data->subject) > $this->module->subjectMaxCharsDisplay)
    {
        $subject .= substr($data->subject,0,$this->module-
>subjectMaxCharsDisplay - strlen($this->module->ellipsis) ).
$this->module->ellipsis . '</a></span>';
    }
    else
    {
        $subject .= $data->subject . '</a></span><span
class="mailbox-msg-brief">'. $subjectSeperator
        .substr(strip_tags($data->text),0,$this->module-
>subjectMaxCharsDisplay - strlen($data->subject) -
strlen($subjectSeperator) - strlen($this->module->ellipsis) );
        if(strlen($data->subject) + strlen($data->text) +
strlen($subjectSeperator) > $this->module->subjectMaxCharsDisplay)
            $subject .= $this->module->ellipsis;
    }
    $subject = preg_replace('/[\n\r]+/', '', $subject);
    $subject .= '</span>';
    ?>
    <tr class="mailbox-item <?php echo $itemCssClass; ?> <?php
if($this->getAction()->getId()!='sent') echo 'mailbox-draggable-
row'; ?>">
        <?php if($this->getAction()->getId()!='sent'): // add
dragdrop handle ?>
            <td width="25" style="width:50px;"><div
class="mailbox-item-wrapper mailbox-drag">&nbsp;</div></td>
        <?php endif; ?>

```

```

<td width="25" style="width:50px;">
    <?php if($this->getAction()->getId()=='sent') : ?>
        <div class="mailbox-item-wrapper">&nbsp;</div>
    <?php else: ?>
        <div class="mailbox-item-wrapper">
            <label class="ui-helper-reset" for="conv_<?php echo
$data->conversation_id; ?>">
                <div class="mailbox-check mailbox-ellipsis">
                    <input class="mailbox-check " id="conv_<?php
echo $data->conversation_id; ?>" type="checkbox" name="convs[]"
value="<?php echo $data->conversation_id; ?>" />
                </div>
            </div>
            </label>
        <?php endif; ?>
    </td>
    <td width="100" style="width:200px;">
        <div class="mailbox-item-wrapper mailbox-from
mailbox-ellipsis"><?php echo $username; ?></div>
    </td>
    <td width="100%" class="mailbox-subject-brief">
        <div class="mailbox-item-wrapper mailbox-item-outer
mailbox-subject"><div class="mailbox-item-inner mailbox-ellipsis
ui-helper-clearfix">
            <?php echo $subject; ?>
        </div></div>
    </td>
    <td width="150" class="mailbox-received">
        <div align="right" class="mailbox-item-wrapper"
style="width:80px">
            <?php if($data->is_replied) : ?>
                <div class="mailbox-replied" title="this message
has been replied to">&nbsp;&nbsp;</div>
            <?php endif; ?>
            <?php echo $received; ?>
        </div>
    </td>
</tr>

```

menu.php

```

<?php
$newMsgs = $this->module->getNewMsgs();
$action = $this->getAction()->getId();

if($this->module->authManager)
{
    $authNew = Yii::app()->user-
>checkAccess("Mailbox.Message.New");
    $authInbox = Yii::app()->user-
>checkAccess("Mailbox.Message.Inbox");
    $authSent = Yii::app()->user-
>checkAccess("Mailbox.Message.Sent");
    $authTrash = Yii::app()->user-
>checkAccess("Mailbox.Message.Trash");
}

```



```

else
{
    $authNew = $this->module->sendMsgs && (!$this->module->readOnly || $this->module->isAdmin());
    $authInbox = ( !$this->module->readOnly || $this->module->isAdmin() );
    $authTrash = $this->module->trashbox && (!$this->module->readOnly || $this->module->isAdmin());
    $authSent = $this->module->sentbox && (!$this->module->readOnly || $this->module->isAdmin());
}
?>
<div class="mailbox-menu ui-helper-clearfix">
    <div class="mailbox-menu-folders ui-helper-clearfix">
        <?php
            if($authInbox):?>
                <div id="mailbox-inbox" class="mailbox-menu-item <?php
echo ($action=='inbox')? 'mailbox-menu-current' : ' ' ; ?>">
                    <a href="<?php echo $this->createUrl('message/inbox'); ?>">Inbox <span class="mailbox-new-msgs"><?php echo $newMsgs? '('. $newMsgs. ')' : null ; ?></span></a>
                </div>
            <?php endif;
            if($authSent) : ?>
                <div id="mailbox-sent" class="mailbox-menu-item <?php
if($action=='sent') echo 'mailbox-menu-current ' ; ?>">
                    <a href="<?php echo $this->createUrl('message/sent'); ?>">Sent Mail</a>
                </div>
            <?php endif;
            if($authTrash) : ?>
                <div id="mailbox-trash" class="mailbox-menu-item <?php
if($action=='trash') echo 'mailbox-menu-current ' ; ?>">
                    <a href="<?php echo $this->createUrl('message/trash'); ?>">Trash </a>
                </div>
            <?php endif; ?>
        </div>
    <?php
    if($authNew) :
        ?>
        <div class="mailbox-menu-newmsg ui-helper-clearfix"
align="center">
            <span><a href="<?php echo $this->createUrl('message/new'); ?>">New Message</a></span>
        </div>
    <?php endif; ?>
</div>

```

compose.php

```

<?php
$this->breadcrumbs=array(
    ucfirst($this->module->id)=>array('inbox'),
    ucfirst($this->getAction()->getId())
);

```

```

$this->renderpartial('_menu');
?>
<div class="mailbox-compose ui-helper-clearfix">

<?php

$this->renderPartial('_flash');

$form=$this->beginWidget('CActiveForm', array(
    'id'=>'message-form',
    'enableAjaxValidation'=>false,
    'htmlOptions'=>array('autocomplete'=>$this-
>createUrl('ajax/auto')),
    'htmlOptions'=>array('enctype'=>'multipart/form-data'),
)); ?>
    <div class="form message-create">

        <div class="mailbox-input-row">
            <div style="float:left">
                <?php echo
CHtml::activeLabelEx($conv,'to'); ?>
            </div>
            <div style="margin-left:80px">
                <?php echo $form-
>textField($conv,'to',array('style'=>'width:100%;','id'=>'message-
to','class'=>'mailbox-input', 'edit'=>$this->module->editToField?
'1' : null)); ?>

                <?php echo $form->error($conv,'to'); ?>

            <?php

                if($this->module->userSupportList)
                {

                    $reps = $this->module-
>getUserSupportList();

                    echo '<select name="ajax[to]"
class="mailbox-support-list" edit="'.(($this->module-
>editToField)? '1' : null).'" >';

                    foreach($reps as $key =>
&$label)
                    {
                        ?>
                        <option type="hidden"
value="<?php echo $key; ?>"><?php echo $label; ?></option>
                        <?php
                        }
                        echo '</select>';
                    }
                }
            </div>
        </div>
    <div class="mailbox-input-row">

```

```

        <div style="float:left">
            <?php echo
CHtml::activeLabelEx($conv,'subject',array('class'=>'mailbox-
label')); ?>
        </div>
        <div class="mailbox-compose-inputwrap"
style="margin-left:80px;">
            <?php echo $form-
>textField($conv,'subject',array('class'=>'mailbox-
input','style'=>'width:100%;','placeholder'=>$this->module-
>defaultSubject)); ?>
            <?php echo $form->error($conv,'subject');
?>
        </div>
        <div style="float:left">
            <?php echo
CHtml::activeLabelEx($msg,'file_foto',array('class'=>'mailbox-
label')); ?>
        </div>
        <div class="mailbox-compose-inputwrap"
style="margin-left:80px;">
            <?php echo $form-
>fileField($msg,'file_foto',array('size'=>30,'maxlength'=>30)); ?>
            <?php echo $form->error($msg,'file_foto');
?>
        </div>
    </div>
    <div class="mailbox-textarea-wrap">
        <?php echo $form-
>textArea($msg,'text',array('cols'=>50,'rows'=>7,
'class'=>'mailbox-message-
input','style'=>'width:100%;','placeholder'=>'Enter message
here...')); ?>
        <?php echo $form->error($msg,'text'); ?>
    </div>
    <div>
        <button class="btn btn-large message-btn-reply">Send
Message</button>
    </div>
</div>
<?php $this->endWidget(); ?><!-- form -->

</div>
<!-- mailbox -->

```

message.php

```

<?php
$this->breadcrumbs=array(
    ucfirst($this->module->id)=>array('mailbox/inbox'),
    'Message',
);

$this->renderPartial('_menu');

$subject = ($conv->subject)? $conv->subject : $this->module-
>defaultSubject;

```

```

if(strlen($subject) > 100)
{
    $subject = substr($subject,0,100);
}

?>
<div class="mailbox-message-list">

<?php
    foreach(Yii::app()->user->getFlashes() as $key => $message) {
        echo '<div class="flash-' . $key . '">' . $message .
"</div>\n";
    }
?>
<div class="mailbox-message-subject mailbox-ellipsis"><?php echo
$subject; ?></div>

<br />
<?php
$first_message=1;
foreach($conv->messages as $msg):
    $sender = $this->module->getUserName($msg->sender_id);
    if(!$sender)
        $sender = $this->module->deletedUser;
    ?>

    <div class="mailbox-message-header">
        <div class="message-sender">
<?php echo ($msg->sender_id == Yii::app()->user->id)? 'You' :
ucfirst($sender);
        echo ($first_message)? ' said' : ' replied'; ?></div>
        <div class="message-date"><?php echo date("Y-m-d H:i
a",$msg->created); ?></div>
        <br />
    </div>
    <div class="mailbox-message-text"><?php echo $msg->text;
?></div>
    <br />
<?php $first_message=0;
endforeach;

if($this->module->authManager)
    $authReply = Yii::app()->user-
>checkAccess("Mailbox.Message.Reply");
else
    $authReply = $this->module->sendMsgs;

if($authReply)
{
    $form=$this->beginWidget('CActiveForm', array(
        'action'=>$this-
>createUrl('message/reply',array('id'=>$_GET['id'])),
        'id'=>'message-reply-form',

```

```

        'enableAjaxValidation'=>false,
    )); ?>
    <div class="mailbox-message-reply ui-helper-clearfix">
        <?php /* echo $form->errorSummary(array($reply,$conv)); */ ?>
        <?php echo $form->error($reply,'text'); ?>
        <div class="mailbox-textarea-wrap ui-helper-clearfix">
            <textarea name="text" cols="50" rows="7"
placeholder="Reply here..."></textarea>
        </div>
        <input type="submit" class="btn btn-large mailbox-input"
value="Send Reply" />
    </div>

<?php $this->endWidget();
}
?>
<?php
$this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'tbl-mailbox-message-grid',
    'dataProvider'=>$msg->search(),
    'filter'=>$msg,
    'columns'=>array(
        array(
            'header'=>'Attachment',
            'type'=>'raw',
            'value'=>'CHtml::link($data->file_foto,
"http://" . $_SERVER["SERVER_NAME"] . Yii::app()->request->baseUrl
. "/images/foto/" . $data->file_foto)',
        ),
    ),
)); ?>
</div>

```

c. Controller

MessageController.php

```

<?php

class MessageController extends Controller
{
    public $defaultAction = 'inbox';

    public $buttons = array( // Ex output: {count} messages have
been {value}

        'default'=>array('delete'=>'deleted','read'=>'marked
read','unread'=>'marked unread'),
        'trash'=>array('delete'=>'permanently
deleted','restore'=>'restored')
    );

    public function filters()
    {
        if($this->module->authManager=='rights') {
            return array(

```

```

        'rights', // perform access control for
        CRUD operations
    );
    }
    else return array();
}

public function behaviors()
{
    return array(
        'ButtonAction'=>array(
            'class'=>'mailbox.behaviors.ButtonActionBehavior',
            'controller'=>$this,
            'module'=>$this->module,
            'buttons'=>$this->buttons,
            'arclass'=>'Mailbox',
        )
    );
}

public function actionInbox($ajax=null)
{
    $this->module->registerConfig($this->getAction()-
>getId());
    $cs = $this->module->getClientScript();
    $cs->registerScriptFile($this->module-
>getAssetsUrl().'/js/mailbox.js',CClientScript::POS_END);
    // $js = '$("#mailbox-list").yiiMailboxList('.$this-
>module->getOptions().');console.log(1)';

    // $cs->registerScript('mailbox-
js',$js,CClientScript::POS_READY);

    if(isset($_POST['convs']))
    {
        $this->buttonAction('inbox');
    }
    $dataProvider = new CActiveDataProvider(
Mailbox::model()->inbox($this->module->getUserId()) );
    if(isset($ajax))
    {
        $this-
>renderPartial('_mailbox',array('dataProvider'=>$dataProvider));
    }
    else{
        if(!isset($_GET['Mailbox_sort']))
            $_GET['Mailbox_sort'] = 'modified.desc';

        $this-
>render('mailbox',array('dataProvider'=>$dataProvider));
    }
}

public function actionSent()
{
    // auth manager

```

```

        if(!$this->module->authManager && (!$this->module-
>sentbox || ($this->module->readOnly && !$this->module-
>isAdmin())) ) )
            $this->redirect(array('message/inbox'));

        $this->module->registerConfig($this->getAction()-
>getId());

        $this->module->getClientScript()-
>registerScriptFile($this->module-
>getAssetsUrl().'/js/jquery.colors.js');
        $this->module->getClientScript()-
>registerScriptFile($this->module-
>getAssetsUrl().'/js/mailbox.js',CClientScript::POS_END);
        if(isset($_POST['convs']))
        {
            $this->buttonAction('sent');
        }
        $dataProvider = new CActiveDataProvider(
Message::model()->sent($this->module->getUserId()) );
        $this-
>render('mailbox',array('dataProvider'=>$dataProvider));
    }

    public function actionTrash($ajax=null)
    {
        // auth manager
        if(!$this->module->authManager && (!$this->module-
>trashbox || ($this->module->readOnly && !$this->module-
>isAdmin())) ) )
            $this->redirect(array('message/inbox'));

        $this->module->registerConfig($this->getAction()-
>getId());

        $this->module->getClientScript()-
>registerScriptFile($this->module-
>getAssetsUrl().'/js/jquery.colors.js');
        $this->module->getClientScript()-
>registerScriptFile($this->module-
>getAssetsUrl().'/js/mailbox.js',CClientScript::POS_END);
        if(isset($_POST['convs']))
        {
            $this->buttonAction('trash','trash');
        }
        $period =& $this->module->recyclePeriod;
        Yii::app()->user->setFlash('notice', "Messages in the
trash are deleted within {$period} days.");
        $dataProvider = new CActiveDataProvider(
Mailbox::model()->trash($this->module->getUserId()) );
        if(isset($ajax))
            $this-
>renderPartial('_mailbox',array('dataProvider'=>$dataProvider));
        else{

```

```

        $this->render('mailbox',array('dataProvider'=>$dataProvider));
    }

    public function actionNew()
    {
        $this->module->registerConfig($this->getAction()->getId());
        $cs = $this->module->getClientScript();
        $cs->registerScriptFile($this->module->getAssetsUrl().'/js/compose.js');
        $cs->registerScriptFile($this->module->getAssetsUrl().'/js/jquery.combobox.contacts.js');
        $js = '$(".mailbox-compose").yiiMailboxCompose("$.this->module->getOptions().");";';
        $cs->registerScript('mailbox-js',$js,CClientScript::POS_READY);
        if(!$this->module->authManager && (!$this->module->sendMsgs || ($this->module->readOnly && !$this->module->isAdmin()) ))
            $this->redirect(array('message/inbox'));

        if(isset($_POST['Mailbox']['to']))
        {
            $t = time();
            $conv = new Mailbox();
            $conv->subject = ($_POST['Mailbox']['subject'])?
            $_POST['Mailbox']['subject'] : $this->module->defaultSubject;
            $conv->to = $_POST['Mailbox']['to'];
            $conv->initiator_id = $this->module->getUserId();

            // Check if username exist
            if(strlen($_POST['Mailbox']['to'])>1)
                $conv->interlocutor_id = $this->module->getUserId($_POST['Mailbox']['to']);
            else
                $conv->interlocutor_id = 0;
            // ...if not check if To field is user id
            if(!$conv->interlocutor_id)
            {
                if($_POST['Mailbox']['to'] && ($this->module->allowLookupById || $this->module->isAdmin()))
                    $username = $this->module->getUserName($_POST['Mailbox']['to']);
                    if(@$username) {
                        $conv->interlocutor_id =
                        $_POST['Mailbox']['to'];
                        $conv->to = $username;
                    }
                else {
                    // possible that javascript was off
                    and user selected from the userSupportList drop down.
                    if(isset($_POST['ajax']['to']) && $this->module->getUserId($_POST['ajax']['to'])) {

```



```

$conv->to =
$_POST['ajax']['to'];
$conv->initiator_id = $this-
>module->getUserId($_POST['ajax']['to']);
}
else
$conv->addError('to', 'User not
found?');
}
}

if($conv->interlocutor_id && $conv->initiator_id
== $conv->interlocutor_id) {
$conv->addError('to', "Can't send message
to self!");
}

if(!$this->module->isAdmin() && $conv-
>interlocutor_id == $this->module->newsUserId){
$conv->addError('to', "User not found?");
}

// check user-to-user perms
if(!$conv->hasErrors() && !$this->module-
>userToUser && !$this->module->isAdmin())
{
if(!$this->module->isAdmin($conv->to))
$conv->addError('to', "Invalid
user!");
}

$conv->modified = $t;
$conv->bm_read = Mailbox::INITIATOR_FLAG;
if($this->module->isAdmin())
$msg = new Message('admin');
else
$msg = new Message('user');
$msg->text = $_POST['Message']['text'];
$validate = $conv-
>validate(array('text'),false); // html purify
$msg->created = $t;
$msg->sender_id = $conv->initiator_id;
$msg->recipient_id = $conv->interlocutor_id;
$msg->file_foto =
CUuploadedFile::getInstance($msg, 'file_foto');
if($this->module->checksums) {
$msg->crc64 = Message::crc64($msg->text);
// 64bit INT
}
else
$msg->crc64 = 0;
// Validate
$validate = $conv->validate(null,false); //
don't clear errors
$validate = $msg->validate() && $validate;

```

```

        if($validate)
        {
            $conv->save();
            $msg->conversation_id = $conv-
>conversation_id;
            $msg->save();
            if(strlen($msg->file_foto)>0)
            $msg->file_foto->saveAs(Yii::app()->basePath
.'../../images/foto/' . $msg->file_foto);
            Yii::app()->user->setFlash('success',
"Message has been sent!");
            $this->redirect(array('message/inbox'));
        }
        else
        {
            Yii::app()->user->setFlash('error', "Error
sending message!");
        }
    }
    else{
        $conv = new Mailbox();
        if(isset($_GET['id']))
            $conv->to = $this->module-
>getUserName($_GET['id']);
        elseif(isset($_GET['to']))
            $conv->to = $_GET['to'];
        else
            $conv->to = '';
        $msg = new Message();
    }
    $this-
>render('compose',array('conv'=>$conv,'msg'=>$msg));
}

public function actionReply()
{
    if(!$this->module->authManager && (!$this->module-
>sendMsgs || ($this->module->readOnly && !$this->module-
>isAdmin()) ))
        $this->redirect(array('message/inbox'));

    $this->module->registerConfig($this->getAction()-
>getId());

    if($this->module->isAdmin())
        $reply = new Message('admin');
    else
        $reply = new Message('user');

    $conv = Mailbox::conversation($_GET['id']);

    if(isset($_POST['text']))
    {
        $reply->text = $_POST['text'];
        $validate = $conv-
>validate(array('text'),false); // html purify

```

```

        $reply->conversation_id = $conv-
>conversation_id;
        if($conv->initiator_id != Yii::app()->user->id)
        {
            $reply->recipient_id = $conv-
>initiator_id;
            $conv->bm_read = $conv->bm_read &
~Mailbox::INITIATOR_FLAG;
        }
        else {
            $reply->recipient_id = $conv-
>interlocutor_id;
            $conv->bm_read = $conv->bm_read &
~Mailbox::INTERLOCUTOR_FLAG;
        }

        $reply->sender_id = $this->module->getUserId();

        $reply->created = time();
        $conv->modified = $reply->created;

        $reply->crc64 = Message::crc64($reply->text);

        $conv->bm_deleted = 0; // restore message
        $conv->interlocutor_del = 0;
        $conv->initiator_del = 0;

        $validate = $reply->validate();
        $validate = $conv->validate() && $validate;

        if($validate)
        {
            $conv->save();
            $reply->save();
            Yii::app()->user->setFlash('success',
"Message sent!");
            $this->redirect(array('message/inbox'));
        }
        else{
            Yii::app()->user->setFlash('error', "Error
sending message!");

            $this-
>render('message',array('conv'=>$conv, 'reply'=>$reply));
        }
    }

    public function actionView()
    {
        $this->module->registerConfig($this->getAction()-
>getId());
        $cs = $this->module->getClientScript();
        $cs->registerScriptFile($this->module-
>getAssetsUrl().' /js/message.js');
    }
}

```

```

        $js = '$(".mailbox-message-
list").yiiMailboxMessage('.$this->module->getOptions().");";
        $cs->registerScript('mailbox-
js',$js,CClientScript::POS_READY);

        $conv = Mailbox::conversation($_GET['id']);

        $conv->markRead($this->module->getUserId());
        $reply = new Message;
        $this->render('message',array('conv'=>$conv,
'reply'=>$reply));
    }
}

```

8. Users

a. Model

User.php

```
<?php
```

```

class User extends CActiveRecord
{
    const STATUS_NOACTIVE=0;
    const STATUS_ACTIVE=1;
    const STATUS_BANNED=-1;

    //TODO: Delete for next version (backward compatibility)
    const STATUS_BANED=-1;

    /**
     * The followings are the available columns in table
     'users':
     * @var integer $id
     * @var string $username
     * @var string $password
     * @var string $email
     * @var string $activkey
     * @var integer $createtime
     * @var integer $lastvisit
     * @var integer $superuser
     * @var integer $status
     * @var timestamp $create_at
     * @var timestamp $lastvisit_at
     */

    /**
     * Returns the static model of the specified AR class.
     * @return CActiveRecord the static model class
     */
    public static function model($className=__CLASS__)
    {
        return parent::model($className);
    }

    /**

```

```

        * @return string the associated database table name
        */
public function tableName()
{
    return Yii::app()->getModule('user')->tableUsers;
}

/**
 * @return array validation rules for model attributes.
 */
public function rules()
{
    // NOTE: you should only define rules for those
attributes that
    // will receive user inputs.CConsoleApplication
    return ((get_class(Yii::app())=='CConsoleApplication'
|| (get_class(Yii::app())!='CConsoleApplication' && Yii::app()-
>getModule('user')->isAdmin()))?array(
        array('username', 'length', 'max'=>30, 'min' =>
3,'message' => UserModule::t("Incorrect username (length between 3
and 20 characters).")),
        array('password', 'length', 'max'=>128, 'min' =>
4,'message' => UserModule::t("Incorrect password (minimal length 4
symbols).")),
        array('email', 'email'),
        array('username', 'unique', 'message' =>
UserModule::t("This user's name already exists.")),
        array('email', 'unique', 'message' =>
UserModule::t("This user's email address already exists.")),
        array('username', 'match', 'pattern' => '/^[A-
Za-z0-9_]+$|/u','message' => UserModule::t("Incorrect symbols (A-
z0-9).")),
        array('status', 'in',
'range'=>array(self::STATUS_NOACTIVE,self::STATUS_ACTIVE,self::STA
TUS_BANNED)),
        array('superuser', 'in', 'range'=>array(0,1)),
        array('create_at', 'default', 'value' => date('Y-m-d
H:i:s'), 'setOnEmpty' => true, 'on' => 'insert'),
        array('lastvisit_at', 'default', 'value' => '0000-00-
00 00:00:00', 'setOnEmpty' => true, 'on' => 'insert'),
        array('username, email, superuser, status',
'required'),
        array('superuser, status', 'numerical',
'integerOnly'=>true),
        array('id, username, password, email, activkey,
create_at, lastvisit_at, superuser, status', 'safe',
'on'=>'search'),
        ):(Yii::app()->user->id==$this->id)?array(
            array('username, email', 'required'),
            array('username', 'length', 'max'=>30, 'min' =>
3,'message' => UserModule::t("Incorrect username (length between 3
and 20 characters).")),
            array('email', 'email'),
            array('username', 'unique', 'message' =>
UserModule::t("This user's name already exists.")),

```

```

        array('username', 'match', 'pattern' => '/^[A-
Za-z0-9_]+$/', 'message' => UserModule::t("Incorrect symbols (A-
z0-9).")),
        array('email', 'unique', 'message' =>
UserModule::t("This user's email address already exists.")),
        ):array());
    }

    /**
     * @return array relational rules.
     */
    public function relations()
    {
        $relations = Yii::app()->getModule('user')->relations;
        if (!isset($relations['profile']))
            $relations['profile'] = array(self::HAS_ONE,
'Profile', 'user_id');
        return $relations;
    }

    /**
     * @return array customized attribute labels (name=>label)
     */
    public function attributeLabels()
    {
        return array(
            'id' => UserModule::t("Id"),
            'username' => UserModule::t("username"),
            'password' => UserModule::t("password"),
            'verifyPassword' => UserModule::t("Retype
Password"),
            'email' => UserModule::t("E-mail"),
            'verifyCode' => UserModule::t("Verification
Code"),
            'activkey' => UserModule::t("activation key"),
            'createtime' => UserModule::t("Registration
date"),
            'create_at' => UserModule::t("Registration
date"),

            'lastvisit_at' => UserModule::t("Last visit"),
            'superuser' => UserModule::t("Superuser"),
            'status' => UserModule::t("Status"),
        );
    }

    public function scopes()
    {
        return array(
            'active' => array(
                'condition' => 'status='.self::STATUS_ACTIVE,
            ),
            'notactive' => array(
                'condition' => 'status='.self::STATUS_NOACTIVE,
            ),
            'banned' => array(

```

```

        'condition'=>'status='.self::STATUS_BANNED,
    ),
    'superuser'=>array(
        'condition'=>'superuser=1',
    ),
    'notsafe'=>array(
        'select' => 'id, username, password, email,
activkey, create_at, lastvisit_at, superuser, status',
    ),
    );
}

    public function defaultScope()
    {
        return CMap::mergeArray(Yii::app()->getModule('user')->defaultScope,array(
            'alias'=>'user',
            'select' => 'user.id, user.username, user.email,
user.create_at, user.lastvisit_at, user.superuser, user.status',
        ));
    }

    public static function itemAlias($type,$code=NULL) {
        $_items = array(
            'UserStatus' => array(
                self::STATUS_NOACTIVE =>
UserModule::t('Not active'),
                self::STATUS_ACTIVE =>
UserModule::t('Active'),
                self::STATUS_BANNED =>
UserModule::t('Banned'),
            ),
            'AdminStatus' => array(
                '0' => UserModule::t('No'),
                '1' => UserModule::t('Yes'),
            ),
        );
        if (isset($code))
            return isset($_items[$type][$code]) ?
$_items[$type][$code] : false;
        else
            return isset($_items[$type]) ? $_items[$type] :
false;
    }

/**
 * Retrieves a list of models based on the current
search/filter conditions.
 * @return CActiveDataProvider the data provider that can
return the models based on the search/filter conditions.
 */
    public function search()
    {
        // Warning: Please modify the following code to remove
attributes that
        // should not be searched.

```

```

$criteria=new CDbCriteria;

$criteria->compare('id',$this->id);
$criteria->compare('username',$this->username,true);
$criteria->compare('password',$this->password);
$criteria->compare('email',$this->email,true);
$criteria->compare('activkey',$this->activkey);
$criteria->compare('create_at',$this->create_at);
$criteria->compare('lastvisit_at',$this->lastvisit_at);
$criteria->compare('superuser',$this->superuser);
$criteria->compare('status',$this->status);

return new CActiveDataProvider(get_class($this), array(
    'criteria'=>$criteria,
    'pagination'=>array(
        'pageSize'=>Yii::app()->getModule('user')->user_page_size,
    ),
));
}

public function getCreatetime() {
    return strtotime($this->create_at);
}

public function setCreatetime($value) {
    $this->create_at=date('Y-m-d H:i:s',$value);
}

public function getLastvisit() {
    return strtotime($this->lastvisit_at);
}

public function setLastvisit($value) {
    $this->lastvisit_at=date('Y-m-d H:i:s',$value);
}

public function getFullName() {
    return $this->username;
}

public function getSuggest($q) {
    $c = new CDbCriteria();
    $c->addSearchCondition('username', $q, true, 'OR');
    $c->addSearchCondition('email', $q, true, 'OR');
    return $this->findAll($c);
}
}

```

b. View

index.php

```

<?php
$this->breadcrumbs=array(
    UserModule::t('Users')=>array('/user'),
    UserModule::t('Manage'),
);

```



```

$this->menu=array(
    array('label'=>UserModule::t('Create User'),
        'url'=>array('create')),
    //array('label'=>UserModule::t('Manage Users'),
        'url'=>array('admin')),
    // array('label'=>UserModule::t('Manage Profile Field'),
        'url'=>array('profileField/admin')),
    //array('label'=>UserModule::t('List User'),
        'url'=>array('/user')),
);

Yii::app()->clientScript->registerScript('search', "
$( '.search-button' ).click(function() {
    $( '.search-form' ).toggle();
    return false;
});
$( '.search-form form' ).submit(function() {
    $.fn.yiiGridView.update('user-grid', {
        data: $(this).serialize()
    });
    return false;
});
");

?>
<h1><?php echo UserModule::t("Manage Users"); ?></h1>

<p><?php echo UserModule::t("You may optionally enter a comparison
operator (<b>&lt;/b>, <b>&lt;=

```

profile.php

```

<?php $this->pageTitle=Yii::app()->name . ' -
'.UserModule::t("Profile");
$this->breadcrumbs=array(
    UserModule::t("Profile"),
);
$this->menu=array(
    ((UserModule::isAdmin()))
        ?array('label'=>UserModule::t('Daftar Users'),
            'url'=>array('/user/admin'))
        :array()),

```

```

        //array('label'=>UserModule::t('List User'),
'url'=>array('/user')),
        array('label'=>UserModule::t('Ubah Data'),
'url'=>array('edit')),
        array('label'=>UserModule::t('Ganti Password'),
'url'=>array('changepassword')),
        array('label'=>UserModule::t('Logout'),
'url'=>array('/user/logout')),
    );
?>
<h1><?php echo UserModule::t('Your profile'); ?></h1>

<?php if(Yii::app()->user->hasFlash('profileMessage')): ?>
<div class="success">
    <?php echo Yii::app()->user->getFlash('profileMessage'); ?>
</div>
<?php endif; ?>
<table class="dataGrid">
    <tr>
        <th class="label"><?php echo CHtml::encode($model->getAttributeLabel('NIM/NIP')); ?></th>
        <td><?php echo CHtml::encode($model->username); ?></td>
    </tr>
    <?php
        $profileFields=ProfileField::model()->forOwner()->sort()->findAll();
        if ($profileFields) {
            foreach($profileFields as $field) {
                //echo "<pre>"; print_r($profile); die();
                ?>
                <tr>
                    <th class="label"><?php echo
CHtml::encode(UserModule::t($field->title)); ?></th>
                    <td><?php echo (($field->widgetView($profile))?$field->widgetView($profile):CHtml::encode(($field->range)?Profile::range($field->range,$profile->getAttribute($field->varname)):$profile->getAttribute($field->varname)))); ?></td>
                </tr>
                <?php
                    }// $profile->getAttribute($field->varname)
                }
            ?>
            <tr>
                <th class="label"><?php echo CHtml::encode($model->getAttributeLabel('email')); ?></th>
                <td><?php echo CHtml::encode($model->email); ?></td>
            </tr>
            <tr>
                <th class="label"><?php echo CHtml::encode($model->getAttributeLabel('Tanggal Daftar')); ?></th>
                <td><?php echo $model->create_at; ?></td>
            </tr>
            <tr>
                <th class="label"><?php echo CHtml::encode($model->getAttributeLabel('Terakhir Masuk')); ?></th>

```

```

        <td><?php echo $model->lastvisit_at; ?></td>
    </tr>
    <tr>
        <th class="label"><?php echo CHtml::encode($model-
>getAttributeLabel('Status')); ?></th>
        <td><?php echo
CHtml::encode(User::itemAlias("UserStatus",$model->status));
?></td>
    </tr>
</table>

```

c. Controller

UserController.php

```

<?php

class UserController extends Controller
{
    /**
     * @var CActiveRecord the currently loaded data model
instance.
     */
    private $_model;

    /**
     * @return array action filters
     */
    public function filters()
    {
        return CMap::mergeArray(parent::filters(),array(
            //'rights',
            'accessControl', // perform access control for
CRUD operations
            'postOnly + delete', // we only allow deletion
via POST request
        ));
    }

    /**
     * Specifies the access control rules.
     * This method is used by the 'accessControl' filter.
     * @return array access control rules
     */
    public function accessRules()
    {
        return array(
            array('allow', // allow admin user to perform
'admin' and 'delete' actions
                'actions'=>array('admin','delete','create','update','view'),
                'hak
aksesess'=>array('SuperAdmin','Staff','Kaprodi'),
            ),
            array('deny', // deny all users
                'users'=>array('*'),

```

```

        ),
    );
}

/**
 * Displays a particular model.
 */
public function actionView()
{
    $model = $this->loadModel();
    $this->render('view', array(
        'model' => $model,
    ));
}

/**
 * Lists all models.
 */
public function actionIndex()
{
    $dataProvider = new CActiveDataProvider('User', array(
        'criteria' => array(
            'condition' => 'status>'.User::STATUS_BANNED,
        ),
        'pagination' => array(
            'pageSize' => Yii::app()->controller-
>module->user_page_size,
        ),
    ));

    $this->render('index', array(
        'dataProvider' => $dataProvider,
    ));
}

public function actionUpdate()
{
    $model = $this->loadModel();
    $profile = $model->profile;
    $this->performAjaxValidation(array($model, $profile));
    if (isset($_POST['User']))
    {
        $model->attributes = $_POST['User'];
        $profile->attributes = $_POST['Profile'];

        if ($model->validate() && $profile->validate()) {
            $old_password = User::model()->notsafe()-
>findByPk($model->id);
            if ($old_password->password != $model-
>password) {
                $model->password = Yii::app()-
>controller->module->encrypting($model->password);
                $model->activkey = Yii::app()-
>controller->module->encrypting(microtime().$model->password);
            }
        }
    }
}

```

```

        $model->save();
        $profile->save();
        $this->redirect(array('view','id'=>$model-
>id));
    } else $profile->validate();
}

$this->render('update',array(
    'model'=>$model,
    'profile'=>$profile,
));
}

public function actionDelete()
{
    if(Yii::app()->request->isPostRequest)
    {
        // we only allow deletion via POST request
        $model = $this->loadModel();
        $profile = Profile::model()->findByPk($model-
>id);

        $profile->delete();
        $model->delete();
        // if AJAX request (triggered by deletion via
admin grid view), we should not redirect the browser
        if(!isset($_POST['ajax']))
            $this->redirect(array('/user/admin'));
    }
    else
        throw new CHttpException(400,'Invalid request.
Please do not repeat this request again.');
```

```

    }

    protected function performAjaxValidation($validate)
    {
        if(isset($_POST['ajax']) && $_POST['ajax']=='user-form')
        {
            echo CActiveForm::validate($validate);
            Yii::app()->end();
        }
    }

    /**
     * Returns the data model based on the primary key given in
the GET variable.
     * If the data model is not found, an HTTP exception will be
raised.
     */
    public function loadModel()
    {
        if($this->_model===null)
        {
            if(isset($_GET['id']))
                $this->_model=User::model()->notsafe()-
>findByPk($_GET['id']);
            if($this->_model===null)

```

```

        throw new CHttpException(404, 'The
requested page does not exist.');
```

```

    }
    return $this->_model;
}

/**
 * Returns the data model based on the primary key given in
the GET variable.
 * If the data model is not found, an HTTP exception will be
raised.
 * @param integer the primary key value. Defaults to null,
meaning using the 'id' GET variable
 */
public function loadUser($id=null)
{
    if($this->_model===null)
    {
        if($id!==null || isset($_GET['id']))
            $this->_model=User::model()-
>findByPk($id!==null ? $id : $_GET['id']);
        if($this->_model===null)
            throw new CHttpException(404, 'The
requested page does not exist.');
```

```

    }
    return $this->_model;
}
}

```

KUESIONER



SURAT KETERANGAN VALIDASI

Yang bertanda tangan di bawah ini:

Nama : Kuswari Hernawati, M.Kom.
NIP : 19760414 200501 2 002
Instansi : FMIPA, Universitas Negeri Yogyakarta

telah memvalidasi instrumen kuesioner dari Tugas Akhir Skripsi dengan judul:

“APLIKASI BIMBINGAN SKRIPSI ONLINE JURUSAN PENDIDIKAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM UNIVERSITAS
NEGERI YOGYAKARTA” yang oleh:

Nama : Muhammad Jaka P.
NIM : 10305141032
Prodi : Matematika
Jurusan : Pendidikan Matematika

Setelah memperhatikan instrumen kuesioner tersebut, maka instrumen kuesioner layak digunakan untuk penelitian dengan beberapa revisi yang telah disampaikan kepada peneliti.

Demikian surat keterangan ini dibuat untuk digunakan semestinya.

Yogyakarta, ... Juni 2014

Validator

Kuswari Hernawati, M.Kom.

NIP. 19760414 200501 2 002



SURAT KETERANGAN REVIEW

Yang bertanda tangan di bawah ini:

Nama : Kuswari Hernawati, M. Kom.
NIP : 19760414 200501 2 002
Instansi : FMIPA, Universitas Negeri Yogyakarta

telah me-review aplikasi dari Tugas Akhir Skripsi dengan judul:

“APLIKASI BIMBINGAN SKRIPSI *ONLINE* JURUSAN PENDIDIKAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM UNIVERSITAS
NEGERI YOGYAKARTA” yang disusun oleh:

Nama : Muhammad Jaka P.
NIM : 10305141032
Prodi : Matematika
Jurusan : Pendidikan Matematika

Setelah me-review aplikasi tersebut, maka aplikasi ini layak digunakan untuk penelitian dengan beberapa revisi yang telah disampaikan kepada peneliti.

Demikian surat keterangan ini dibuat untuk digunakan semestinya.

Yogyakarta, Juli 2014

Reviewer

Kuswari Hernawati, M. Kom

NIP. 19760414 200501 2 002

Kuesioner Penelitian

Hari, tanggal :
Nama :
NIM :
Tipe Pengguna : Mahasiswa

Judul Penelitian

Aplikasi Bimbingan Skripsi *Online* Jurusan Pendidikan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta

Tujuan Penelitian

1. Mengevaluasi kekurangan/hal yang perlu diperbaiki dari sistem aplikasi yang telah dibuat.
2. Mengetahui seberapa besar aplikasi dapat membantu proses pembuatan tugas akhir.

Petunjuk:

1. Mohon dengan hormat bantuan dan kesedian Bapak/Ibu/Saudara(i) untuk menjawab seluruh pertanyaan yang ada dalam kuesioner sesuai dengan pengetahuan anda.
2. Berilah tanda (x) pada pilihan jawaban yang paling sesuai pada pertanyaan yang ada dalam kuesioner.

Keterangan pilihan jawaban:

STS : Sangat Tidak Setuju

TS : Tidak Setuju

S : Setuju

SS : Sangat Setuju

No	Keterangan	STS	TS	S	SS
Correctness					
1	Menu yang tersedia memenuhi kebutuhan			X	
2	Fitur-fitur yang tersedia di Aplikasi Bimbingan Skripsi <i>Online</i> membantu proses pembuatan skripsi				X
3	Butir-butir informasi pada menu Judul memenuhi kebutuhan			X	
4	Menu Upload Proposal memudahkan mahasiswa untuk mengajukan proposal skripsi kepada Ketua Program Studi tanpa harus bertatap muka			X	
5	Menu Pesan efektif untuk membantu proses bimbingan skripsi			X	
Reliability					
6	Pencarian judul skripsi pada menu Judul (judul skripsi) menampilkan hasil yang sesuai dengan kata kunci yang dimasukkan			X	
7	Pencarian data dosen pada menu Dosen (data dosen) akurat			X	
8	Pesan yang dibuat pada menu Pesan sesuai dengan pesan yang terkirim				X
9	Data diri yang ditampilkan sesuai dengan data yang di-input saat mendaftar				X
Integrity					
10	Data yang anda input aman dari orang yang tidak berhak mengakses			X	
Usability					
11	Struktur menu dan fungsi dalam sistem mudah dipahami				X
12	Instruksi atau petunjuk yang tersedia membantu menggunakan aplikasi			X	
13	Bahasa yang digunakan mudah dipahami				X
14	Tulisan di semua halaman dapat terbaca dengan jelas				X
15	Tampilan Aplikasi Bimbingan Skripsi <i>Online</i> menarik			X	

Kesimpulan

Berdasarkan pengujian pada Aplikasi Bimbingan Skripsi *Online* dan jawaban kuesioner di atas, maka Aplikasi Bimbingan Skripsi *Online* tersebut dinyatakan **LAYAK / ~~TIDAK LAYAK~~**)* untuk digunakan.

)* coret yang tidak perlu.

Saran:

Menu pencarian apakah lebih baik bisa menampilkan kata2 yg dimasukkan baik itu kataanya saja atau seluruh kata yang dimasukkan dan hasil tampilannya bisa menunjukkan kata yang dimasukkan pada bagian mana

Yogyakarta, 9 Juni 2014

Responden



(M. Bostan Nur R.)

No	Keterangan	STS	TS	S	SS
Correctness					
1	Menu yang tersedia memenuhi kebutuhan			X	
2	Fitur-fitur yang tersedia di Aplikasi Bimbingan Skripsi <i>Online</i> membantu proses pembuatan skripsi				X
3	Butir-butir informasi pada menu Judul memenuhi kebutuhan			X	
4	Menu Upload Proposal memudahkan mahasiswa untuk mengajukan proposal skripsi kepada Ketua Program Studi tanpa harus bertatap muka				X
5	Menu Pesan efektif untuk membantu proses bimbingan skripsi			X	
Reliability					
6	Pencarian judul skripsi pada menu Judul (judul skripsi) menampilkan hasil yang sesuai dengan kata kunci yang dimasukkan				X
7	Pencarian data dosen pada menu Dosen (data dosen) akurat			X	
8	Pesan yang dibuat pada menu Pesan sesuai dengan pesan yang terkirim				X
9	Data diri yang ditampilkan sesuai dengan data yang di-input saat mendaftar			X	
Integrity					
10	Data yang anda input aman dari orang yang tidak berhak mengakses				X
Usability					
11	Struktur menu dan fungsi dalam sistem mudah dipahami			X	
12	Instruksi atau petunjuk yang tersedia membantu menggunakan aplikasi				X
13	Bahasa yang digunakan mudah dipahami				X
14	Tulisan di semua halaman dapat terbaca dengan jelas				X
15	Tampilan Aplikasi Bimbingan Skripsi <i>Online</i> menarik			X	

Kesimpulan

Berdasarkan pengujian pada Aplikasi Bimbingan Skripsi *Online* dan jawaban kuesioner di atas, maka Aplikasi Bimbingan Skripsi *Online* tersebut dinyatakan **LAYAK / ~~TIDAK LAYAK~~**)* untuk digunakan.

)* *coret yang tidak perlu.*


Saran:

dan detail dosen

Pada menu profile apakah baik ditambahkan ~~contact~~
person, foto.

Yogyakarta, 9 Juni 2014

Responden


(Rizal Arroyid)

No	Keterangan	STS	TS	S	SS
Correctness					
1	Menu yang tersedia memenuhi kebutuhan			X	
2	Fitur-fitur yang tersedia di Aplikasi Bimbingan Skripsi <i>Online</i> membantu proses pembuatan skripsi			X	
3	Butir-butir informasi pada menu Judul memenuhi kebutuhan			X	
4	Menu Upload Proposal memudahkan mahasiswa untuk mengajukan proposal skripsi kepada Ketua Program Studi tanpa harus bertatap muka				X
5	Menu Pesan efektif untuk membantu proses bimbingan skripsi			X	
Reliability					
6	Pencarian judul skripsi pada menu Judul (judul skripsi) menampilkan hasil yang sesuai dengan kata kunci yang dimasukkan				X
7	Pencarian data dosen pada menu Dosen (data dosen) akurat				X
8	Pesan yang dibuat pada menu Pesan sesuai dengan pesan yang terkirim				X
9	Data diri yang ditampilkan sesuai dengan data yang di-input saat mendaftar				X
Integrity					
10	Data yang anda input aman dari orang yang tidak berhak mengakses			X	
Usability					
11	Struktur menu dan fungsi dalam sistem mudah dipahami				X
12	Instruksi atau petunjuk yang tersedia membantu menggunakan aplikasi				X
13	Bahasa yang digunakan mudah dipahami				X
14	Tulisan di semua halaman dapat terbaca dengan jelas				X
15	Tampilan Aplikasi Bimbingan Skripsi <i>Online</i> menarik			X	

Kesimpulan

Berdasarkan pengujian pada Aplikasi Bimbingan Skripsi *Online* dan jawaban kuesioner di atas, maka Aplikasi Bimbingan Skripsi *Online* tersebut dinyatakan **LAYAK / ~~TIDAK LAYAK~~**)* untuk digunakan.

)* *coret yang tidak perlu.*

Saran:

.....
Profil ditambahkan foto, CP
.....
.....
.....

Yogyakarta, 9 Juni 2014

Responden



(..... Dwi P.)

No	Keterangan	STS	TS	S	SS
Correctness					
1	Menu yang tersedia memenuhi kebutuhan			X	
2	Fitur-fitur yang tersedia di Aplikasi Bimbingan Skripsi <i>Online</i> membantu proses pembuatan skripsi				X
3	Butir-butir informasi pada menu Judul memenuhi kebutuhan			X	
4	Menu Upload Proposal memudahkan mahasiswa untuk mengajukan proposal skripsi kepada Ketua Program Studi tanpa harus bertatap muka				X
5	Menu Pesan efektif untuk membantu proses bimbingan skripsi				X
Reliability					
6	Pencarian judul skripsi pada menu Judul (judul skripsi) menampilkan hasil yang sesuai dengan kata kunci yang dimasukkan				X
7	Pencarian data dosen pada menu Dosen (data dosen) akurat				X
8	Pesan yang dibuat pada menu Pesan sesuai dengan pesan yang terkirim				X
9	Data diri yang ditampilkan sesuai dengan data yang di-input saat mendaftar				X
Integrity					
10	Data yang anda input aman dari orang yang tidak berhak mengakses			X	
Usability					
11	Struktur menu dan fungsi dalam sistem mudah dipahami				X
12	Instruksi atau petunjuk yang tersedia membantu menggunakan aplikasi				X
13	Bahasa yang digunakan mudah dipahami				X
14	Tulisan di semua halaman dapat terbaca dengan jelas				X
15	Tampilan Aplikasi Bimbingan Skripsi <i>Online</i> menarik			X	

Kesimpulan

Berdasarkan pengujian pada Aplikasi Bimbingan Skripsi *Online* dan jawaban kuesioner di atas, maka Aplikasi Bimbingan Skripsi *Online* tersebut dinyatakan **LAYAK / ~~TIDAK LAYAK~~**)* untuk digunakan.

)* *coret yang tidak perlu.*

Saran:

.....

.....

.....

.....

.....

Yogyakarta, 9 Juni 2014

Responden



(ERNI KURNIANINGSIH)

No	Keterangan	STS	TS	S	SS
Correctness					
1	Menu yang tersedia memenuhi kebutuhan				X
2	Fitur-fitur yang tersedia di Aplikasi Bimbingan Skripsi <i>Online</i> membantu proses pembuatan skripsi			X	
3	Butir-butir informasi pada menu Judul memenuhi kebutuhan			X	
4	Menu Upload Proposal memudahkan mahasiswa untuk mengajukan proposal skripsi kepada Ketua Program Studi tanpa harus bertatap muka			X	
5	Menu Pesan efektif untuk membantu proses bimbingan skripsi			X	
Reliability					
6	Pencarian judul skripsi pada menu Judul (judul skripsi) menampilkan hasil yang sesuai dengan kata kunci yang dimasukkan				X
7	Pencarian data dosen pada menu Dosen (data dosen) akurat				X
8	Pesan yang dibuat pada menu Pesan sesuai dengan pesan yang terkirim			X	
9	Data diri yang ditampilkan sesuai dengan data yang di-input saat mendaftar			X	
Integrity					
10	Data yang anda input aman dari orang yang tidak berhak mengakses			X	
Usability					
11	Struktur menu dan fungsi dalam sistem mudah dipahami				X
12	Instruksi atau petunjuk yang tersedia membantu menggunakan aplikasi			X	
13	Bahasa yang digunakan mudah dipahami				X
14	Tulisan di semua halaman dapat terbaca dengan jelas				X
15	Tampilan Aplikasi Bimbingan Skripsi <i>Online</i> menarik		X		

Kesimpulan

Berdasarkan pengujian pada Aplikasi Bimbingan Skripsi *Online* dan jawaban kuesioner di atas, maka Aplikasi Bimbingan Skripsi *Online* tersebut dinyatakan **LAYAK** / ~~**TIDAK LAYAK**~~)* untuk digunakan.

)* coret yang tidak perlu.

Saran:

Tampilan diperbaiki
Hindari & Ruang kosong pada website
Font diperbesar

Yogyakarta, 9 Juni 2014

Responden



(...arif kurniawan...)

Kuesioner Penelitian

Hari, tanggal :
Nama :
NIP :
Tipe Pengguna : Dosen Pembimbing Skripsi

Judul Penelitian

Aplikasi Bimbingan Skripsi *Online* Jurusan Pendidikan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta

Tujuan Penelitian

1. Mengevaluasi kekurangan/hal yang perlu diperbaiki dari sistem aplikasi yang telah dibuat.
2. Mengetahui seberapa besar aplikasi dapat membantu proses pembuatan tugas akhir.

Petunjuk:

1. Mohon dengan hormat bantuan dan kesedian Bapak/Ibu/Saudara(i) untuk menjawab seluruh pertanyaan yang ada dalam kuesioner sesuai dengan pengetahuan anda.
2. Berilah tanda (x) pada pilihan jawaban yang paling sesuai pada pertanyaan yang ada dalam kuesioner.

Keterangan pilihan jawaban:

STS : Sangat Tidak Setuju

TS : Tidak Setuju

S : Setuju

SS : Sangat Setuju

No	Keterangan	STS	TS	S	SS
Correctness					
1	Menu yang tersedia memenuhi kebutuhan			X	
2	Fitur-fitur yang tersedia di Aplikasi Bimbingan Skripsi <i>Online</i> membantu proses pembuatan skripsi				X
3	Menu Dosen (data dosen) diperlukan bagi dosen		X		
4	Menu Pesan efektif untuk membantu proses bimbingan skripsi				X
Reliability					
5	Pencarian judul skripsi pada menu Judul (judul skripsi) menampilkan hasil yang sesuai dengan kata kunci yang dimasukkan				X
6	Pesan yang dibuat pada menu Pesan sesuai dengan pesan yang terkirim				X
7	Data diri yang ditampilkan sesuai dengan data yang di-input saat mendaftar				X
Integrity					
8	Data yang di-input aman dari orang yang tidak berhak mengakses			X	
Usability					
9	Struktur menu dan fungsi dalam sistem mudah dipahami				X
10	Instruksi atau petunjuk yang tersedia membantu menggunakan aplikasi				X
11	Bahasa yang digunakan mudah dipahami				X
12	Tulisan di semua halaman dapat terbaca dengan jelas				X
13	Tampilan Aplikasi Bimbingan Skripsi <i>Online</i> menarik			X	

Kesimpulan

Berdasarkan pengujian pada Aplikasi Bimbingan Skripsi *Online* dan jawaban kuesioner di atas, maka Aplikasi Bimbingan Skripsi *Online* tersebut dinyatakan **LAYAK** / ~~TIDAK LAYAK~~)* untuk digunakan.

)* coret yang tidak perlu.

Saran:

* Kalau bisa ditambah fasilitas
pada menu login, yaitu data GUGI
skripsi yg da GUGI + status selesai / GUGI

Yogyakarta, Juni 2014

Responden



(.....) Alletheps

No	Keterangan	STS	TS	S	SS
Correctness					
1	Menu yang tersedia memenuhi kebutuhan		X		
2	Fitur-fitur yang tersedia di Aplikasi Bimbingan Skripsi <i>Online</i> membantu proses pembuatan skripsi			X	
3	Menu Dosen (data dosen) diperlukan bagi dosen		X		
4	Menu Pesan efektif untuk membantu proses bimbingan skripsi			X	
Reliability					
5	Pencarian judul skripsi pada menu Judul (judul skripsi) menampilkan hasil yang sesuai dengan kata kunci yang dimasukkan			X	
6	Pesan yang dibuat pada menu Pesan sesuai dengan pesan yang terkirim			X	
7	Data diri yang ditampilkan sesuai dengan data yang di-input saat mendaftar			X	
Integrity					
8	Data yang di-input aman dari orang yang tidak berhak mengakses			X	
Usability					
9	Struktur menu dan fungsi dalam sistem mudah dipahami			X	
10	Instruksi atau petunjuk yang tersedia membantu menggunakan aplikasi			X	
11	Bahasa yang digunakan mudah dipahami			X	
12	Tulisan di semua halaman dapat terbaca dengan jelas			X	
13	Tampilan Aplikasi Bimbingan Skripsi <i>Online</i> menarik			X	

Kesimpulan

Berdasarkan pengujian pada Aplikasi Bimbingan Skripsi *Online* dan jawaban kuesioner di atas, maka Aplikasi Bimbingan Skripsi *Online* tersebut dinyatakan **LAYAK / TIDAK LAYAK**)* untuk digunakan.

)* *coret yang tidak perlu.*

Saran:

.....

.....

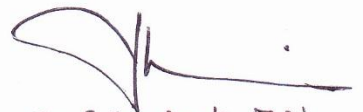
.....

.....

.....

Yogyakarta, Juni 2014

Responden


(Kus. Prihantoso K. M. Si.)

Kuesioner Penelitian

Hari, tanggal :
Nama :
NIP :
Tipe Pengguna : Staf Jurusan

Judul Penelitian

Aplikasi Bimbingan Skripsi *Online* Jurusan Pendidikan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta

Tujuan Penelitian

1. Mengevaluasi kekurangan/hal yang perlu diperbaiki dari sistem aplikasi yang telah dibuat.
2. Mengetahui seberapa besar aplikasi dapat membantu proses pembuatan tugas akhir.

Petunjuk:

1. Mohon dengan hormat bantuan dan kesedian Bapak/Ibu/Saudara(i) untuk menjawab seluruh pertanyaan yang ada dalam kuesioner sesuai dengan pengetahuan anda.
2. Berilah tanda (x) pada pilihan jawaban yang paling sesuai pada pertanyaan yang ada dalam kuesioner.

Keterangan pilihan jawaban:

STS : Sangat Tidak Setuju

TS : Tidak Setuju

S : Setuju

SS : Sangat Setuju

No	Keterangan	STS	TS	S	SS
Correctness					
1	Menu yang tersedia memenuhi kebutuhan				X
2	Fitur-fitur yang tersedia di Aplikasi Bimbingan Skripsi <i>Online</i> membantu proses pembuatan skripsi				X
3	Menu Pesan membantu untuk melakukan pemberitahuan kepada pengguna				X
Reliability					
4	Menu Judul (judul skripsi) menampilkan informasi yang sesuai dengan data yang di-input				X
5	Pesan yang dibuat pada menu pesan sesuai dengan pesan yang terkirim				X
6	Role pengguna yang ditampilkan sesuai dengan data <i>role</i> pengguna yang di-input				X
7	Data diri yang ditampilkan sesuai dengan data yang di-input saat mendaftar				X
Integrity					
8	Data yang di-input aman dari orang yang tidak berhak mengakses				X
Usability					
9	Struktur menu dan fungsi dalam sistem mudah dipahami				X
10	Instruksi atau petunjuk yang tersedia membantu menggunakan aplikasi				X
11	Tampilan Aplikasi Bimbingan Skripsi <i>Online</i> menarik			X	
12	Pengelolaan data (tambah, edit, hapus data) pada menu Judul mudah dilakukan				X
13	Pengelolaan data (tambah, edit, hapus data) pada menu Dosen mudah dilakukan				X
14	Menu <i>Role</i> mudah digunakan				X
15	Pengelolaan data (tambah, edit, hapus data) pada menu Users (data pengguna) mudah dilakukan				X

Kesimpulan

Berdasarkan pengujian pada Aplikasi Bimbingan Skripsi *Online* dan jawaban kuesioner di atas, maka Aplikasi Bimbingan Skripsi *Online* tersebut dinyatakan **LAYAK / ~~TIDAK LAYAK~~**)* untuk digunakan.

)* *coret yang tidak perlu.*

Saran:

Untuk tampilan akan lebih menarik jika
full colour sehingga membuat pengguna lebih
tertarik.

Yogyakarta, Juni 2014

Responden



(.....APRIANISA N.F.)

No	Keterangan	STS	TS	S	SS
Correctness					
1	Menu yang tersedia memenuhi kebutuhan			X	
2	Fitur-fitur yang tersedia di Aplikasi Bimbingan Skripsi <i>Online</i> membantu proses pembuatan skripsi			X	
3	Menu Pesan membantu untuk melakukan pemberitahuan kepada pengguna			X	
Reliability					
4	Menu Judul (judul skripsi) menampilkan informasi yang sesuai dengan data yang di-input			X	
5	Pesan yang dibuat pada menu pesan sesuai dengan pesan yang terkirim			X	
6	<i>Role</i> pengguna yang ditampilkan sesuai dengan data <i>role</i> pengguna yang di-input			X	
7	Data diri yang ditampilkan sesuai dengan data yang di-input saat mendaftar			X	
Integrity					
8	Data yang di-input aman dari orang yang tidak berhak mengakses				X
Usability					
9	Struktur menu dan fungsi dalam sistem mudah dipahami			X	
10	Instruksi atau petunjuk yang tersedia membantu menggunakan aplikasi			X	
11	Tampilan Aplikasi Bimbingan Skripsi <i>Online</i> menarik			X	
12	Pengelolaan data (tambah, edit, hapus data) pada menu Judul mudah dilakukan			X	
13	Pengelolaan data (tambah, edit, hapus data) pada menu Dosen mudah dilakukan			X	
14	Menu <i>Role</i> mudah digunakan			X	
15	Pengelolaan data (tambah, edit, hapus data) pada menu Users (data pengguna) mudah dilakukan			X	

Kesimpulan

Berdasarkan pengujian pada Aplikasi Bimbingan Skripsi *Online* dan jawaban kuesioner di atas, maka Aplikasi Bimbingan Skripsi *Online* tersebut dinyatakan **LAYAK / ~~TIDAK LAYAK~~**)* untuk digunakan.

)* *coret yang tidak perlu.*

Saran:

- Mungkin dapat ditambah dalam menu Judul Tambahan Kolom Pengusi (karena pengusi ada 4).

Yogyakarta, Juni 2014

Responden



(Yossi Inti R. S.E...)

Kuesioner Penelitian

Hari, tanggal :
Nama :
NIP :
Tipe Pengguna : Ketua Program Studi

Judul Penelitian

Aplikasi Bimbingan Skripsi *Online* Jurusan Pendidikan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta

Tujuan Penelitian

1. Mengevaluasi kekurangan/hal yang perlu diperbaiki dari sistem aplikasi yang telah dibuat.
2. Mengetahui seberapa besar aplikasi dapat membantu proses pembuatan tugas akhir.

Petunjuk:

1. Mohon dengan hormat bantuan dan kesedian Bapak/Ibu/Saudara(i) untuk menjawab seluruh pertanyaan yang ada dalam kuesioner sesuai dengan pengetahuan anda.
2. Berilah tanda (x) pada pilihan jawaban yang paling sesuai pada pertanyaan yang ada dalam kuesioner.

Keterangan pilihan jawaban:

STS : Sangat Tidak Setuju

TS : Tidak Setuju

S : Setuju

SS : Sangat Setuju

No	Keterangan	STS	TS	S	SS
Correctness					
1	Menu yang tersedia memenuhi kebutuhan		X		
2	Fitur-fitur yang tersedia di Aplikasi Bimbingan Skripsi <i>Online</i> membantu proses pembuatan skripsi			X	
3	Menu Dosen (data dosen) diperlukan bagi Ketua Program Studi				X
4	Fitur download proposal membantu Ketua Program Studi untuk menerima file proposal skripsi dari mahasiswa tanpa harus bertatap muka				X
Reliability					
5	Pencarian judul skripsi pada menu Judul (judul skripsi) menampilkan hasil yang sesuai dengan kata kunci yang dimasukkan				X
6	Menu Judul (judul skripsi) menampilkan informasi yang sesuai dengan data yang di-input				X
Integrity					
7	Data yang di-input aman dari orang yang tidak berhak mengakses				X
Usability					
8	Struktur menu dan fungsi dalam sistem mudah dipahami			X	
9	Instruksi atau petunjuk yang tersedia membantu menggunakan aplikasi			X	
10	Tampilan Aplikasi Bimbingan Skripsi <i>Online</i> menarik		X		
11	Bahasa yang digunakan mudah dipahami			X	
12	Tulisan di semua halaman dapat terbaca dengan jelas			X	
13	Pengelolaan data (tambah, edit, hapus data) pada menu Judul mudah dilakukan			X	

Kesimpulan

Berdasarkan pengujian pada Aplikasi Bimbingan Skripsi *Online* dan jawaban kuesioner di atas, maka Aplikasi Bimbingan Skripsi *Online* tersebut dinyatakan **LAYAK / ~~TIDAK LAYAK~~**)* untuk digunakan.


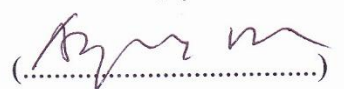
)* coret yang tidak perlu.

Saran:

Dipublikasikan secara periodik
& melalui media

Yogyakarta, Juni 2014

Responden


()

Kuesioner Penelitian

Hari, tanggal :
Nama :
NIP :

Judul Penelitian

Aplikasi Bimbingan Skripsi *Online* Jurusan Pendidikan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta

Tujuan Penelitian

1. Mengevaluasi kekurangan/hal yang perlu diperbaiki dari sistem aplikasi yang telah dibuat.
2. Mengetahui seberapa besar aplikasi dapat membantu proses pembuatan tugas akhir.

Petunjuk:

1. Mohon dengan hormat bantuan dan kesedian Bapak/Ibu/Saudara(i) untuk menjawab seluruh pertanyaan yang ada dalam kuesioner sesuai dengan pengetahuan anda.
2. Berilah tanda (x) pada pilihan jawaban yang paling sesuai pada pertanyaan yang ada dalam kuesioner.

Keterangan pilihan jawaban:

STS : Sangat Tidak Setuju

TS : Tidak Setuju

S : Setuju

SS : Sangat Setuju

No	Keterangan	STS	TS	S	SS
Efficiency					
1	Penggunaan code pada sistem efisien			X	
Maintainability					
2	Pengolahan data skripsi mudah dilakukan			X	
3	Pengolahan data pengguna mudah dilakukan			X	
Testability					
4	Pencarian judul skripsi pada menu Judul (judul skripsi) menampilkan hasil yang sesuai dengan kata kunci yang dimasukkan				X
5	Menu Judul (judul skripsi) menampilkan informasi yang sesuai dengan data yang di-input				X
6	Pesan yang dibuat pada menu pesan sesuai dengan pesan yang terkirim				X
7	File proposal yang di-upload pada menu Upload Proposal muncul pada list download proposal pada menu List Proposal				X
Flexibility					
8	Sistem yang dibuat dengan bahasa PHP dapat dimodifikasi dengan bahasa pemrograman java script			X	
Portability					
9	Sistem dapat ditampilkan/diakses di beberapa hardware yaitu komputer, tablet, dan smartphone				X
10	Sistem dapat ditampilkan/diakses di beberapa browser yaitu mozilla firefox, dan google chrome				X
Reusability					
11	Data yang tersimpan di database dapat diakses melalui di Ms. excel			X	
Interoperability					
12	Proses pendaftaran pengguna (Mahasiswa) berhubungan dengan proses upload proposal			X	

Kesimpulan

Berdasarkan pengujian pada Aplikasi Bimbingan Skripsi *Online* dan jawaban kuesioner di atas, maka Aplikasi Bimbingan Skripsi *Online* tersebut dinyatakan **LAYAK / ~~TIDAK LAYAK~~**)* untuk digunakan.

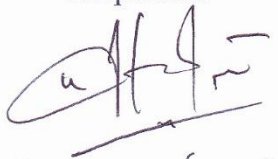
)* *coret yang tidak perlu.*

Saran:

Sebaiknya user dosen dapat melihat daftar mahasiswa bimbingannya pada saat login -
pembatasan pesan / bimbingan khusus untuk mahasiswa bimbingannya sendiri.

Yogyakarta, Juli 2014

Responden


(Kuswari H.)